Riconoscimento e recupero dell'informazione per bioinformatica

Hidden Markov Models

Manuele Bicego

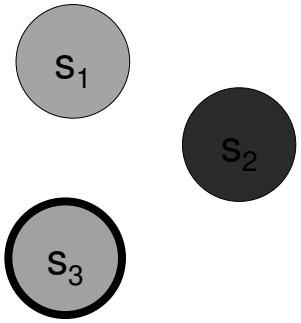
Corso di Laurea in Bioinformatica
Dipartimento di Informatica - Università di Verona

Sommario

- ⇨ Processi e modelli di Markov
- ⇨ Processi e modelli di Markov a stati nascosti (Hidden Markov Models)
- ⇨ Hidden Markov Models per la bioinformatica: profile HMM

2

Processo di Markov (ordine 1)



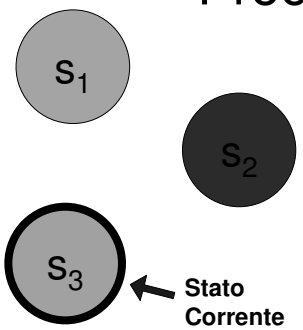
N=3
t=1

- Ha N stati , $s_1,s_2,\ldots,s_N$
- E' caratterizzato da passi discreti, t=1,t=2,…
- La probabilità di partire da un determinato stato è dettata dalla distribuzione:

$\Pi=\{\pi_i\}: \ \pi_i = P(q_1 = s_i)$ con

$1 \le i \le N, \ \pi_i \ge 0$ e $\sum_{i=1}^{N} \pi_i = 1$

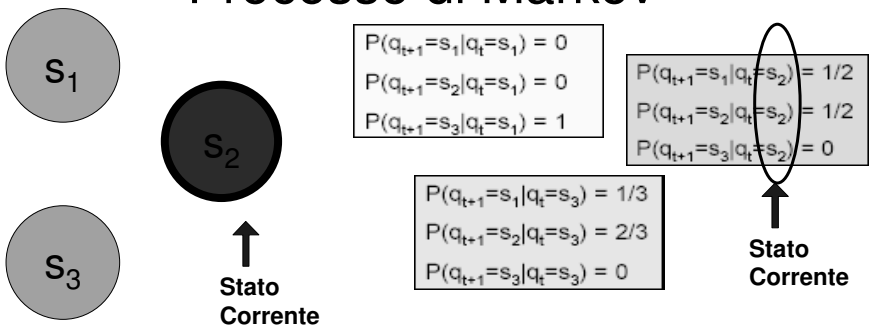
Processo di Markov



N=3
t=1
$q_1=s_3$

- Al t-esimo istante il processo si trova esattamente in uno degli stati a disposizione, indicato dalla variabile q_t
- Nota: $q_t \in \{s_1,s_2,\ldots,s_N\}$
- Ad ogni iterazione, lo stato successivo viene scelto con una determinata probabilità

Processo di Markov



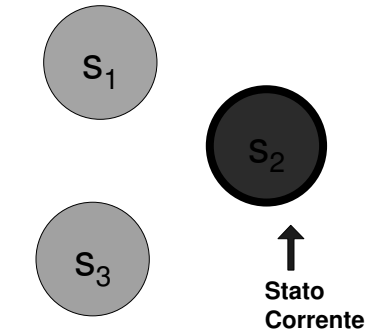
N=3 t=2

$q_2=s_2$

•Tale probabilità è unicamente determinata dallo stato precedente (*markovianetà di primo ordine*):

$$P(q_{t+1}= s_j|q_t= s_i,q_{t-1}=s_k,\ldots,q_1=s_l) = P(q_{t+1}= s_j|q_t= s_i)$$

Processo di Markov



N=3 t=1

$q_2=s_2$

•Definendo:

$a_{i,j} = P(q_{t+1}= s_j \mid q_t = s_i)$

ottengo la matrice NxN

A di *transizione tra stati*, *invariante nel tempo*:

$$A= \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,1} & a_{3,3} \end{bmatrix}$$

Caratteristiche dei processi Markoviani

- Sono processi (discreti) caratterizzati da:
  - Markovianità del primo ordine
  - una distribuzione iniziale
- Conoscendo le caratteristiche di cui sopra, si può esibire un ***modello (probabilistico) di Markov (MM)*** come

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

7

Cosa serve un modello stocastico?

- Modella e riproduce **processi sequenziali**
- Descrive tramite probabilità **le *cause* che portano da uno stato all'altro del sistema**
- In altre parole, più è probabile che dallo stato A si passi allo stato B, più è probabile che **A causi B**

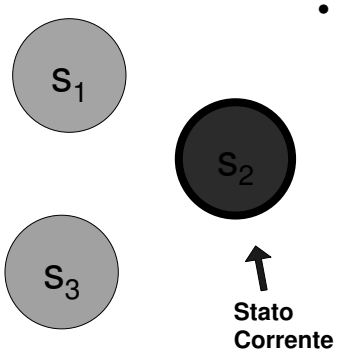
8

Che operazioni si possono eseguire su un modello probabilistico?

- **Addestramento o training**
  - Si costruiscono gli elementi costituenti del modello
- **Inferenze di vario tipo (interrogo il modello):**
  - Probabilità di una sequenza di stati, dato il modello
  - Proprietà invarianti etc.

9

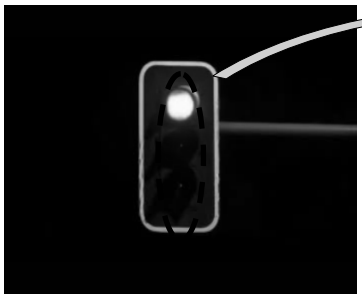
Cosa serve un modello di Markov?

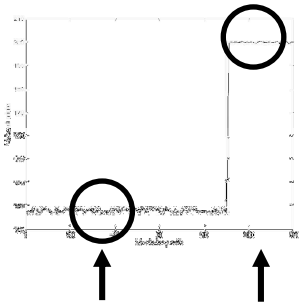


- Modella comportamenti *stocastici Markoviani (di ordine N)* di un sistema in cui gli stati sono:
  - **Espliciti** (riesco a dar loro un nome)
  - **Osservabili** (ho delle osservazioni che univocamente identificano lo stato)

10

Esempio: Semaforo



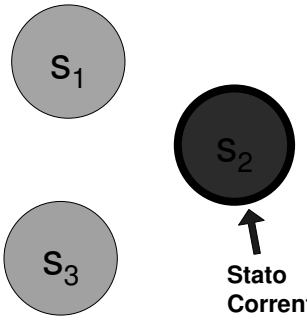


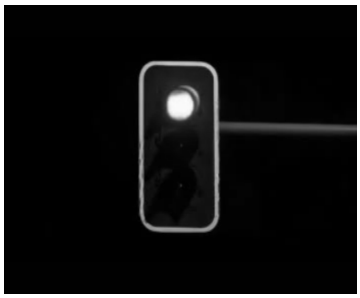
t=1 t=2 …

- E' un sistema di cui gli stati sono:
  - **Espliciti** (le diverse lampade accese)
  - **Osservabili** (i colori delle lampade che osservo con la telecamera)

11

Semaforo – modello addestrato





$\pi =$

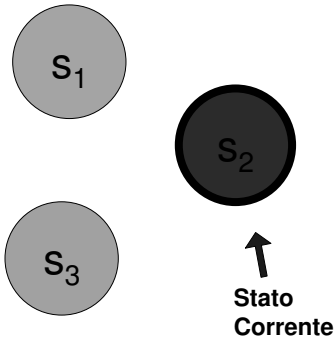
$\pi_1 = 0.33$	$\pi_2 = 0.33$	$\pi_3 = 0.33$

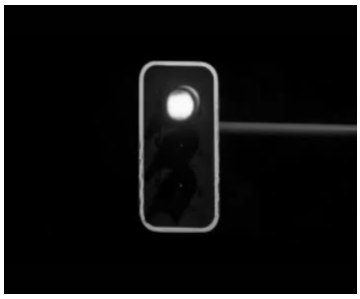
$A =$

$a_{11} = 0.1$	$a_{12} = 0.9$	$a_{13} = 0$
$a_{21} = 0.01$	$a_{22} = 0$	$a_{23} = 0.99$
$a_{31} = 1$	$a_{32} = 0$	$a_{33} = 0$

12

Semaforo – inferenze





$O_2 = < q_2 = s_3, q_1 = s_2 >$

Inferenza: $P(O|\lambda) = P(O)$

$= P(q_2 = s_3, q_1 = s_2) = P(q_2, q_1)$

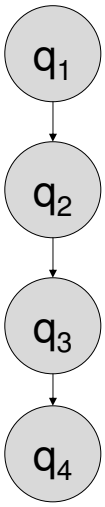
13

Inferenza importante

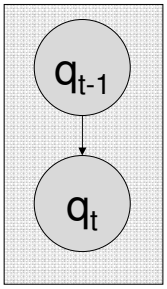
$$
\begin{aligned}
P(q_t, q_{t-1}, \ldots, q_1) &= P(q_t | q_{t-1}, \ldots, q_1)\, P(q_{t-1}, \ldots, q_1) \\
&= P(q_t | q_{t-1})\, P(q_{t-1}, q_{t-2}, \ldots, q_1) \\
&= P(q_t | q_{t-1})\, P(q_{t-1} | q_{t-2})\, P(q_{t-2}, \ldots, q_1) \\
&\ldots \\
&= P(q_t | q_{t-1})\, P(q_{t-1} | q_{t-2}) \ldots P(q_2 | q_1) P(q_1)
\end{aligned}
$$

14

Modello grafico



- La struttura grafica di tale probabilità congiunta si scrive in questa forma, dove

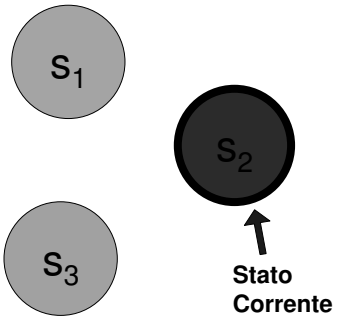


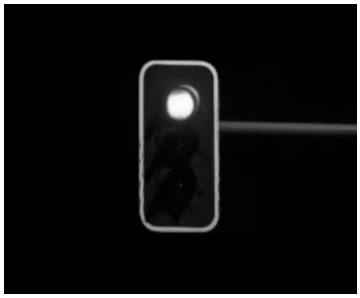
$= P(q_t|q_{t-1})$

$(=P(q_t|q_{t-1},\ldots,q_1)$ ***in questo caso***$)$

15

Semaforo – inferenze, risposta





$P(O|\lambda) = P(O)$

$= P(q_2= s_3, q_1 = s_2)$

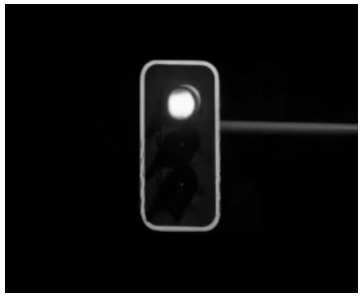
$= P(q_2= s_3|q_1 = s_2)\ P(q_1 = s_2)$

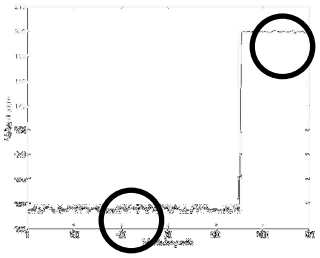
$= \quad 0.99 \quad * \quad 0.33 = 0.326$

16

Limiti dei modelli markoviani

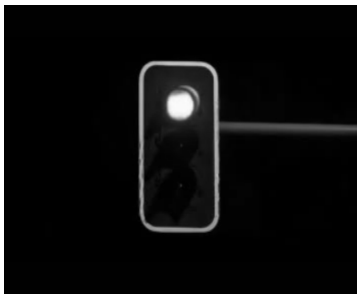
1. *Lo stato è sempre* ***osservabile deterministicamente***, tramite le osservazioni (non c'è rumore).



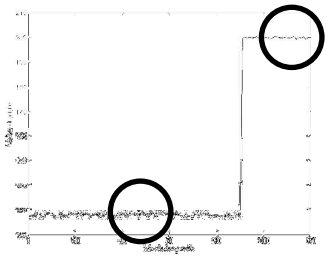


17

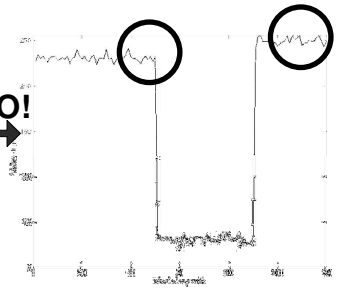
Limiti dei modelli markoviani



OK



NO!

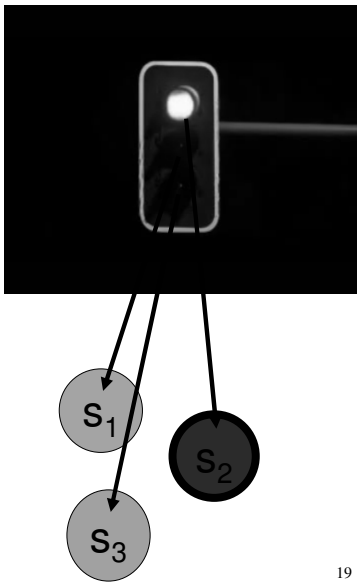


Limiti dei modelli markoviani

2. (Più importante!) Nel caso del semaforo lo stato è **esplicito**, (una particolare configurazione del semaforo), e **valutabile direttamente tramite l'osservazione**

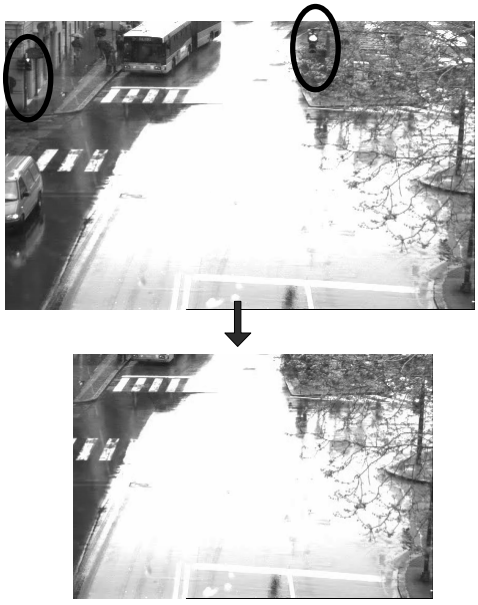
   (lo stato corrisponde al colore del semaforo)

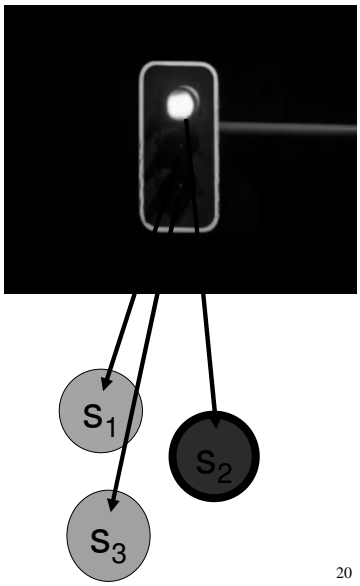
- Non sempre accade così!



19

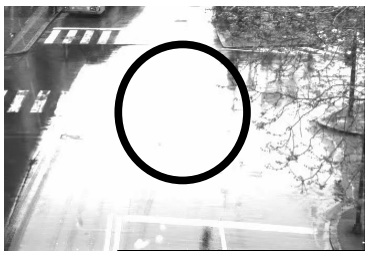
Limiti dei modelli markoviani

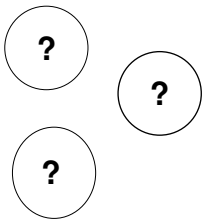




20

Limiti dei modelli markoviani

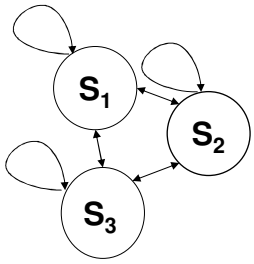




- Osservo il filmato: osservo che c'è un **sistema che evolve**, ma non riesco a capire quali siano gli stati regolatori.

21

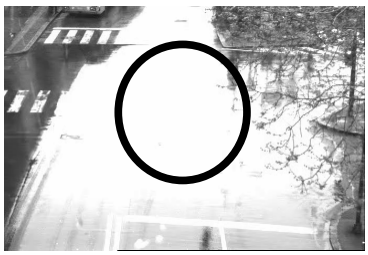
Limiti dei modelli markoviani

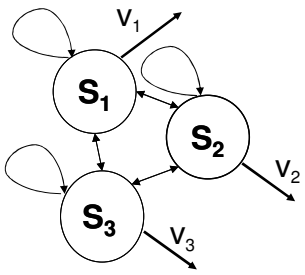


- *Osservo* il filmato: *osservo* che c'è un **sistema che evolve**, ma non riesco a capire quali siano gli stati regolatori.
- Il sistema comunque evolve a **stati**; lo capisco *osservando* il fenomeno (introduco una conoscenza "a priori")

22

Limiti dei modelli markoviani

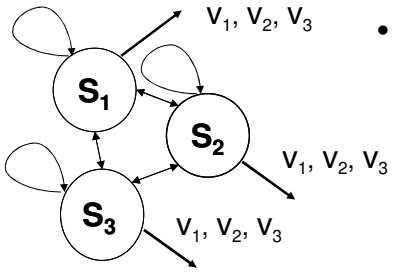




- Meglio: il sistema evolve grazie a degli **stati "nascosti"** (gli stati del semaforo, che però non vedo e di cui ignoro l'esistenza) di cui riesco ad ***osservare*** solo le probabili "conseguenze", ossia i flussi delle auto

23

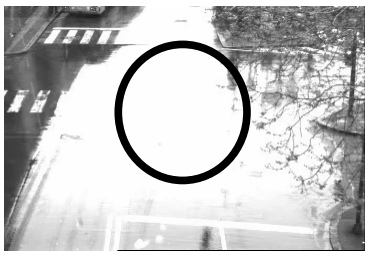
Limiti dei modelli markoviani

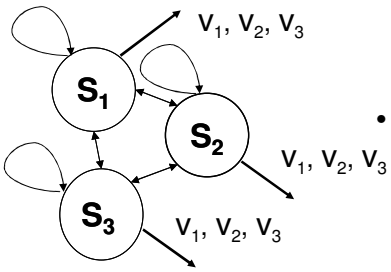


- Rinuncio a dare un nome agli stati, li penso come entità nascoste e *identificabili solo tramite osservazioni* (i flussi delle auto)
- Stabilisco una **relazione tra osservazione e stato nascosto.**

24

Modelli markoviani a stati nascosti (HMM)





- Gli Hidden Markov Model si inseriscono in questo contesto
- Descrivono probabilisticamente la *dinamica di un sistema* rinunciando ad identificarne direttamente le cause
- Gli *stati* sono identificabili solamente tramite le *osservazioni*, in maniera probabilistica.

25

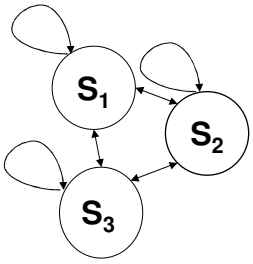
Hidden Markov Model (HMM)

- Classificatore statistico di sequenze, molto utilizzato negli ultimi anni in diversi contesti
- Tale modello può essere inteso come estensione del modello di Markov dal quale differisce per la non osservabilità dei suoi stati
- Suppongo ora di avere per le mani un HMM addestrato, ossia in grado di descrivere un sistema stocastico come descritto sopra...

26

HMM definizione formale

- Un HMM (discreto) è formato da:
  - Un insieme $S=\{s_1,s_2,\ldots,s_N\}$ di stati nascosti;
  - Una matrice di transizione $A=\{a_{ij}\}$, tra stati nascosti 1=<i,j=<N
  - Una distribuzione iniziale sugli stati nascosti $\pi=\{\pi_i\}$,



$\pi=$

π_1= 0.33	π_2= 0.33	π_3= 0.33

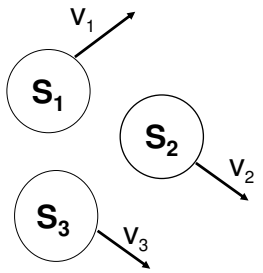
$A=$

a_{11}= 0.1	a_{12}= 0.9	a_{13}=0
a_{21}= 0.01	a_{22}= 0.2	a_{23}= 0.79
a_{31}= 1	a_{32}= 0	a_{33}= 0

27

HMM: definizione formale

- Un insieme $V=\{v_1,v_2,\ldots,v_M\}$ *di simboli d'osservazione;*
- Una distribuzione di probabilità sui simboli d'osservazione $B=\{b_{jk}\}$, *che indica la probabilità di emissione del simbolo* v_k quando lo stato del sistema è s_j.



$B=$

b_{11}=0.8	b_{21}=0.1	b_{31}=0.1
b_{12}= 0.1	b_{22}= 0.8	b_{32}= 0.1
b_{1M}= 0.1	b_{2M}= 0.1	b_{3M}=0.8

28

HMM: definizione formale

- Denotiamo una HMM con una tripla λ=(A, B, π)

$\boldsymbol{\pi}$=

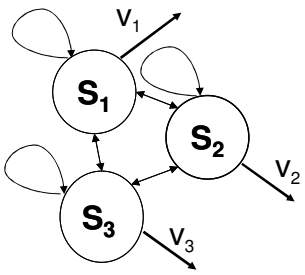
π_1= 0.33	π_2= 0.33	π_3= 0.33

A=

a_{11}= 0.1	a_{12}= 0.9	a_{13}=0
a_{21}= 0.01	a_{22}= 0.2	a_{23}= 0.79
a_{31}= 1	a_{32}= 0	a_{33}= 0

B=

b_{11}=0.8	b_{21}=0.1	b_{31}=0.1
b_{12}= 0.1	b_{22}= 0.8	b_{32}= 0.1
b_{1M}= 0.1	b_{2M}= 0.1	b_{3M}=0.8

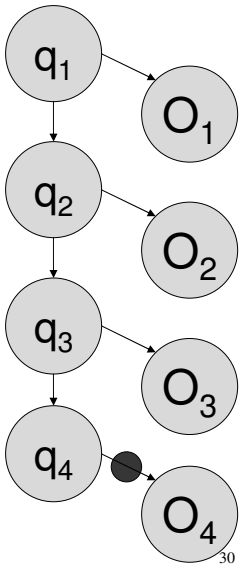


29

Assunzioni sull'osservazione

- Indipendenze condizionali

$$P(O_t=X|q_t=s_j,\ q_{t-1},q_{t-2},\ldots,q_2,q_1,\ O_{t-1},O_{t-2},\ldots,O_2,O_1) = P(O_t=X|q_t=s_j)$$



30

Urn & Ball – An easy example

- ***N*** large urns with ***M*** coloured balls in each
- Urns are the states and balls are the observable events
- Transition matrix for changing between urns
- Each urn has observation probabilities to determine which ball is chosen

31

Urn & Ball – An Example

Urn 1

P(red) = $b_1(1)$
P(blue) = $b_1(2)$
P(green) = $b_1(3)$
P(purple) = $b_1(4)$
…

Urn 2

P(red) = $b_2(1)$
P(blue) = $b_2(2)$
P(green) = $b_2(3)$
P(purple) = $b_2(4)$
…

Urn 3

P(red) = $b_3(1)$
P(blue) = $b_3(2)$
P(green) = $b_3(3)$
P(purple) = $b_3(4)$
…

32

Urn & Ball – An Example

- Initial probability to determine first urn
- At each time interval:
  - Transition probability determines the urn
  - Observation probability determines the ball
  - Ball colour added to observed event sequence and returned to urn
- Transition probability dependent on previous urn

33

Example Sequence Creation using Urn Ball

1. From π, 1st urn = Urn 1
2. Using $b_1(k)$, 1st ball = Red
3. From a_{1j}, 2nd urn = Urn 3 etc…

- Get observation sequence
  - Red, Blue, Purple, Yellow, Blue, Blue
- From state sequence
  - Urn1, Urn 3, Urn 3, Urn 1, Urn, 2, Urn 1

34

HMM basic problems

1. *Evaluation*: given an observation string **O** and a model λ, compute P(**O**|λ)
2. *Decoding*: given an observation string **O** and a model λ, compute the optimal state sequence $Q_1,...,Q_T$ generating the sequence **O**
3. *Training*: given a set of observation sequences $\{\mathbf{O}_l\}$, determine the best model λ, i.e. the model for which $P(\{\mathbf{O}_l\}|\lambda)$ is maximized

35

HMM evaluation

"Given an observation string **O** and a model λ, compute P(**O**|λ)"

First Option: given the sequence $O=O_1,...,O_T$

$$P(O \mid \lambda) = \sum_{\text{All sequences } Q_1,...,Q_T} \pi_{Q_1} b_{Q_1}(O_1) a_{Q_1Q_2} b_{Q_2}(O_2) a_{Q_2Q_3} ...$$

Problem: too high complexity!!

Solution: Forward-Backward procedure

36

HMM evaluation

Forward – Backward procedure.

- Recursively compute

  $\alpha_t(i)=P(O_1 \ldots O_t, Q_t=S_i | \lambda)$

  $\beta_t(i)=P(O_{t+1} \ldots O_T, q_t=S_i | \lambda)$

- $P(O | \lambda)$ is then computed as

$$P(O | \lambda) = \sum_{i=1}^{N} \alpha_t(i)\beta_t(i) \qquad \forall t$$

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

37

HMM decoding

Given an observation string **O**= $O_1..O_T$ and a model λ, compute the optimal state sequence $Q_1,...,Q_T$ generating the sequence **O**

Problem: what is “optimality”?

38

HMM decoding

- First possibility: choose the states Q_t that are "individually" most likely

Drawbacks:

1. The optimal sequence could be not valid if there are some transitions with zero probabilities ($a_{ij} = 0$ for some i and j)

Optimizing the "states", not the "sequence of states"

39

HMM decoding

- Common choice: find the "single" best state sequence (path)
- Maximization of $P(Q|O, \lambda)$

Formal solution to this problem: dynamic programming technique, called "Viterbi algorithm"

40

HMM training

given a set of observation sequences $\{\mathbf{O}_l\}$, determine the best model λ, i.e. the model for which $P(\{\mathbf{O}_l\}|\lambda)$ is maximized

PROBLEM: no analytical solution

Baum-Welch re-estimation technique

41

HMM training

- Baum Welch re-estimation procedure:
  - variation of the well known Expectation-Maximization (EM) algorithm
  - local optimizer
  - maximize log-likelihood of the model w.r.t. data

$$\lambda_{opt} = \arg\max \log P(\{\mathbf{O}_l\}|\lambda)$$

42

HMM training

1. Initialize the model $\lambda' = (\mathbf{A}_0, \mathbf{B}_0, \boldsymbol{\pi}_0)$
2. the current model is $\lambda = \lambda'$
3. While (stop conditions met)

   E-step: use current model λ to compute sufficient statistics (E-STEP)

   M-step: re-estimate the parameters using the sufficient statistics, obtaining the new model $\lambda' = (\mathbf{A}_{new}, \mathbf{B}_{new}, \boldsymbol{\pi}_{new})$
4. the final model is λ'.

43

HMM training

- Baum showed that at each step

$P(\{\mathbf{O}_l\}|\lambda') > P(\{\mathbf{O}_l\}|\lambda)$

- Usual stop conditions:
  - after a fixed number of iterations (maybe 20)
  - when the likelihood converges

44

HMM training

Fundamental issue:

- Baum-Welch is a gradient-descent optimization technique (local optimizer)
- the log-likelihood is highly multimodal

- initialization of parameters can crucially affect the convergence of the algorithm

45

HMM training initialization

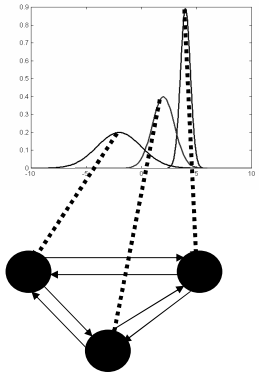
- Discrete HMM: several random initialization + picking up the best one
- Gaussian HMM: awkward problem!

One possible solution:

⇨ "unroll" the sequence

⇨ fit with a mixture of Gaussians

⇨ Each Gaussian used for each state



GMM clustering

46

Riassumendo: HMM per la classificazione

- Dato un problema a C classi:
- Training:
  - utilizzare tutte le sequenze della classe Ck per addestrare l'HMM che rappresenta quella classe
- Testing:
  - dato un oggetto sconosciuto, si classifica con la regola di Bayes
  - si calcola la likelihood dell'oggetto con ogni HMM (eventualmente si moltiplica per il prior)
  - si assegna l'oggetto alla classe la cui posterior è massima

47