

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: pair HMM e profile HMM

Hidden Markov Models

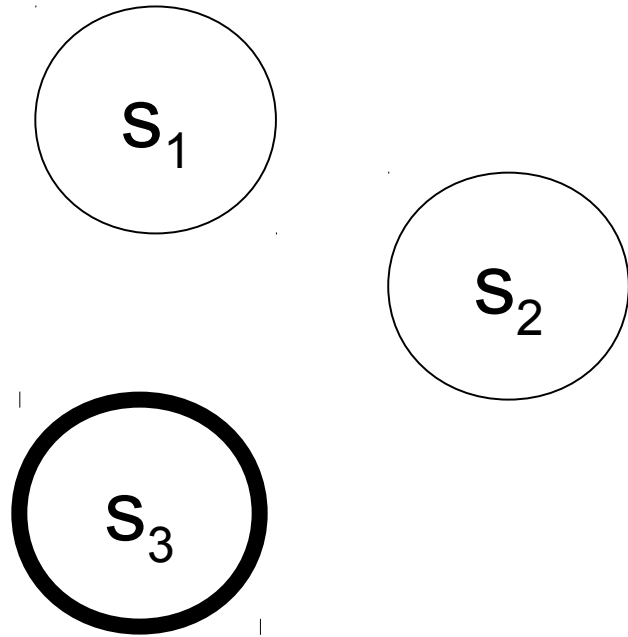
- ⇒ Approccio molto diffuso per la modellazione di dati sequenziali
- ⇒ Caratteristiche principali:
 - ⇒ La capacità intrinseca di modellare l'evoluzione sequenziale
 - ⇒ La presenza di un algoritmo di training veloce ed efficace (basato su Expectation-Maximization – EM)
 - ⇒ La chiara interpretazione Bayesiana
 - ⇒ Se usati bene, questi modelli funzionano molto bene in un sacco di applicazioni
 - ⇒ Molto utilizzati in bioinformatica per la modellazione di sequenze:
 - ⇒ Profile HMM, pair HMM

Hidden Markov Models

⇒ Un HMM può essere visto come un'estensione di un modello di Markov (o processo di Markov)

vediamo cos'è un processo di Markov

Processi di Markov (di ordine 1)



$N=3$
 $t=1$

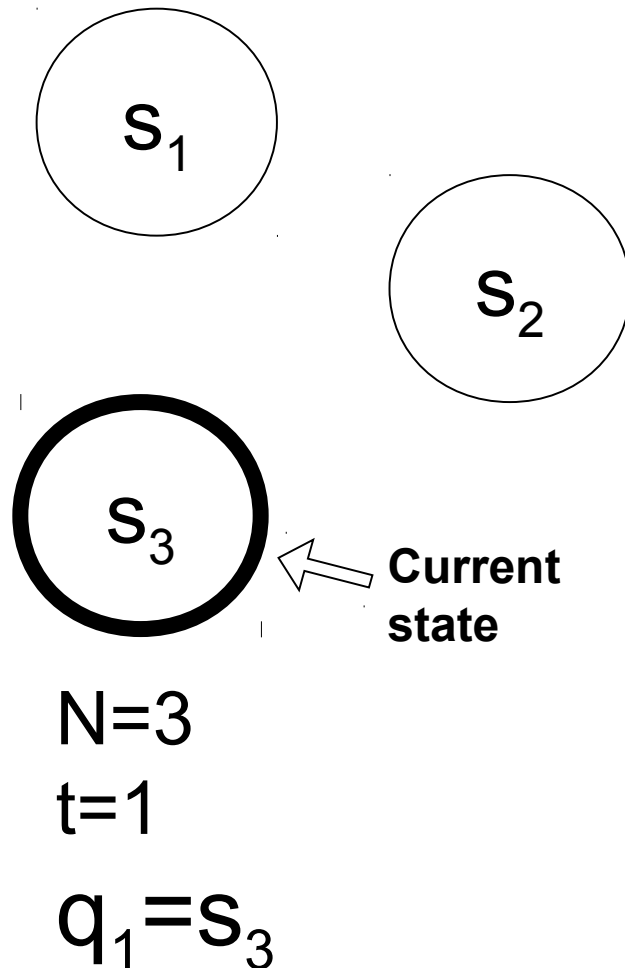
- Processo che modella un sistema che evolve secondo N stati , $s_1, s_2, \dots, s_N$
- Il sistema evolve con passi discreti $t=1, t=2, \dots$
- La probabilità di partire da uno stato è data da

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

$$1 \leq i \leq N, \pi_i \geq 0 \text{ e}$$

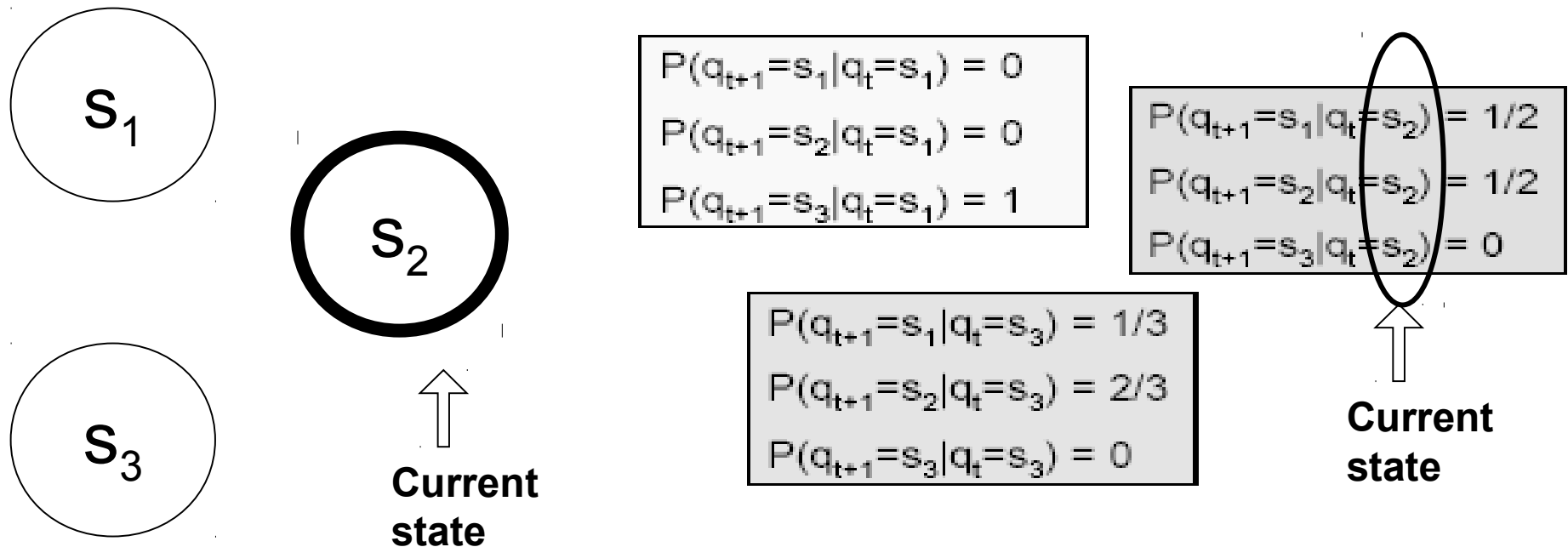
$$\sum_{i=1}^N \pi_i = 1$$

Processi di Markov (di ordine 1)



- Al t -esimo istante il processo si trova in uno degli stati possibili, indicati come q_t
- Nota: $q_t \in \{s_1, s_2, \dots, s_N\}$
- Ad ogni iterazione, il prossimo stato è scelto con una certa probabilità

Processi di Markov (di ordine 1)



N=3 t=2

$q_2=s_2$

•Tale probabilità è determinata unicamente dallo stato precedente (*markovianità del primo ordine*)

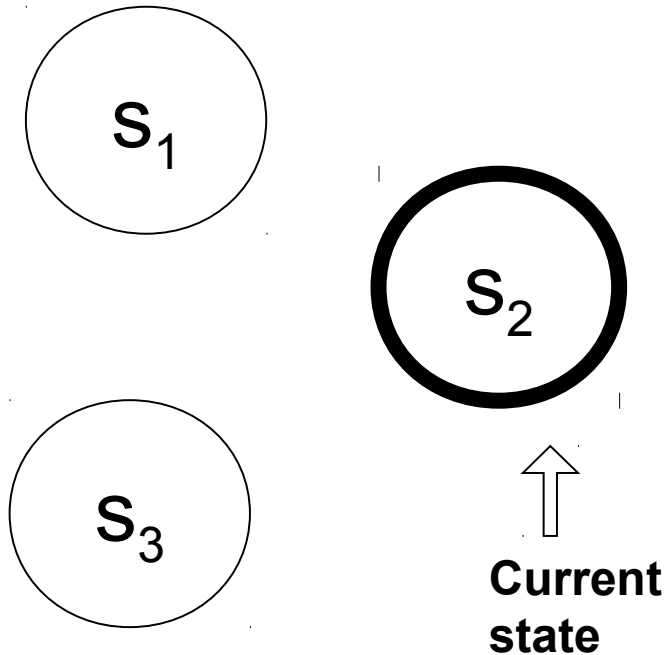
$$P(q_{t+1}=s_j|q_t=s_i, q_{t-1}=s_k, \dots, q_1=s_l) = P(q_{t+1}=s_j|q_t=s_i)$$

Processi di Markov (di ordine 1)

•Definendo:

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

possiamo ottenere la matrice A ($N \times N$) delle transizioni tra gli stati (*time invariant*):



$N=3 \ t=1$

$q_2 = s_2$

$A =$

$a_{1,1}$	$a_{1,2}$	$a_{1,3}$
$a_{2,1}$	$a_{2,2}$	$a_{2,3}$
$a_{3,1}$	$a_{3,1}$	$a_{3,3}$

Caratteristiche dei processi markoviani

- ⇒ Sono processi discreti, caratterizzati da:
 - ⇒ Markovianità del primo ordine (codificata nella matrice A)
 - ⇒ Una distribuzione iniziale
- ⇒ Possiamo definire un modello (probabilistico) di Markov come
$$\lambda = (A, \pi)$$

Cosa può “fare” un modello di Markov?

- ⇒ Può modellare e riprodurre un **processo sequenziale**
- ⇒ Può descrivere, attraverso le probabilità, le cause che portano un sistema da uno stato ad un altro
- ⇒ In altre parole, più è alta la probabilità di passare dallo stato A allo stato B, più è alta la probabilità che **A causi B**

Che operazioni possono essere fatte su un modello di Markov?

⇒ Addestramento:

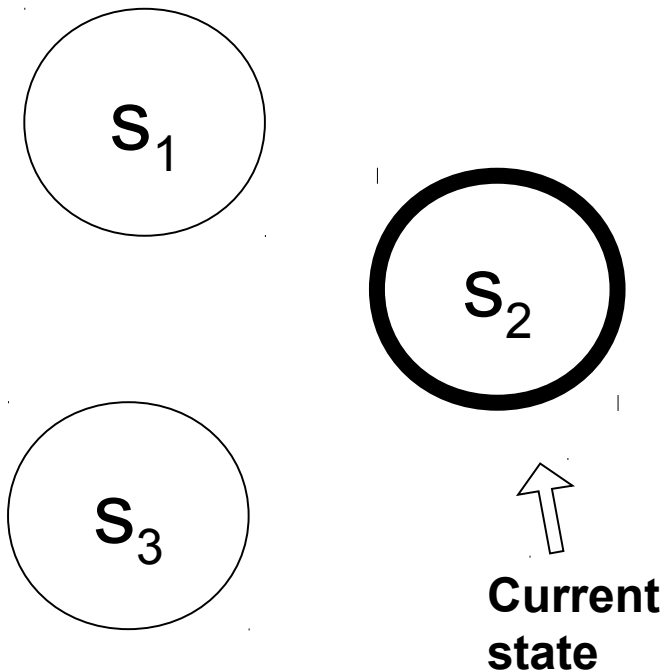
⇒ Inferire i parametri del modello

⇒ Inferenza (“Chiedere qualcosa al modello”)

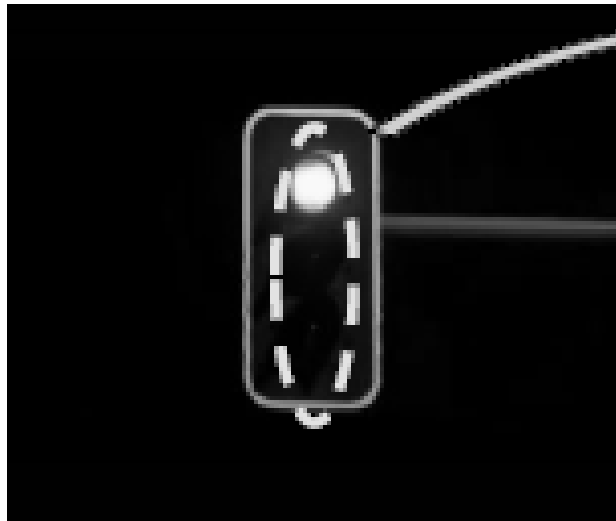
⇒ Esempio: la probabilità di passare attraverso una sequenza di stati

Riassumendo

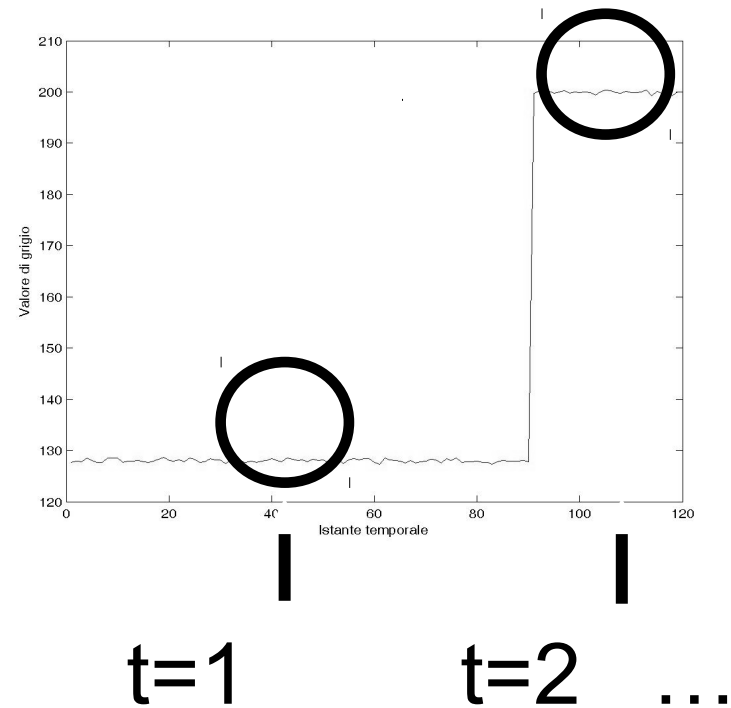
- Un modello di Markov può descrivere un comportamento stocastico markoviano (di ordine 1) di un sistema i cui stati sono:
 - **Espliciti** (posso dare un nome ad ogni stato)
 - **Osservabili** (ci sono delle osservazioni che identificano univocamente lo stato)



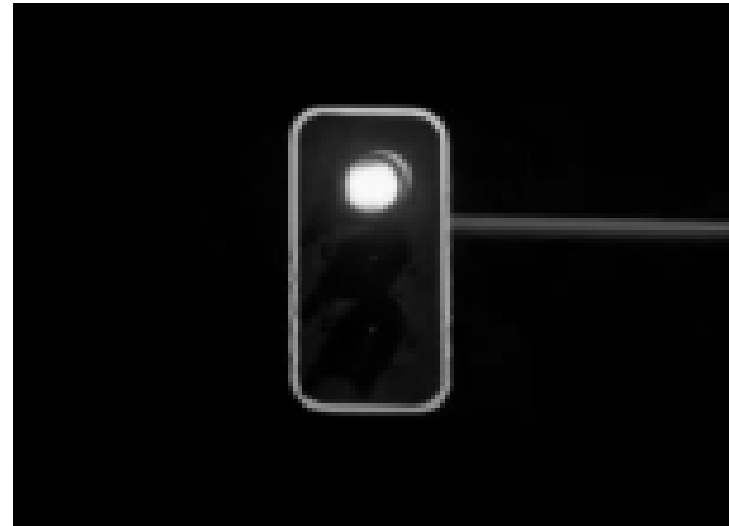
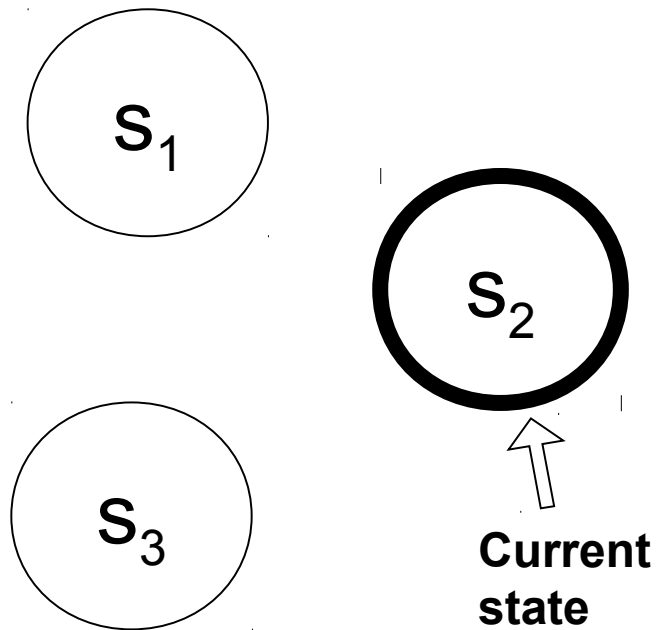
Esempio: il semaforo



- Un sistema dove gli stati sono:
 - **Espliciti** (le lampade)
 - **Osservabili** (Posso utilizzare una telecamera per “vedere” il colore del semaforo)



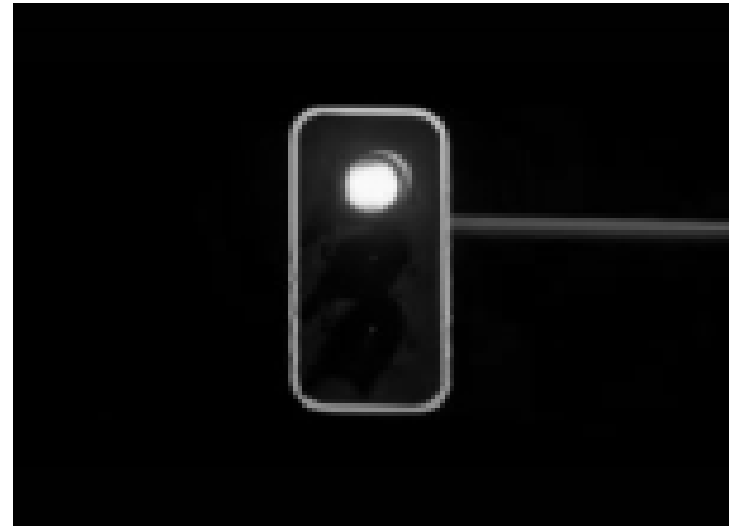
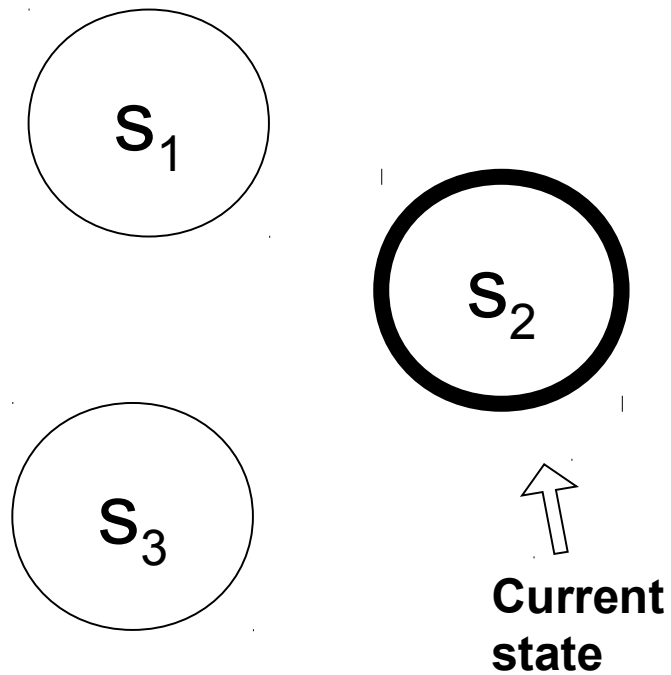
Modello del semaforo addestrato



$$\pi = \begin{matrix} \pi_1 = 0.33 & \pi_2 = 0.33 & \pi_3 = 0.33 \end{matrix}$$

$$A = \begin{cases} a_{11} = 0.1 & a_{12} = 0.9 & a_{13} = 0 \\ a_{21} = 0.01 & a_{22} = 0 & a_{23} = 0.99 \\ a_{31} = 1 & a_{32} = 0 & a_{33} = 0 \end{cases}$$

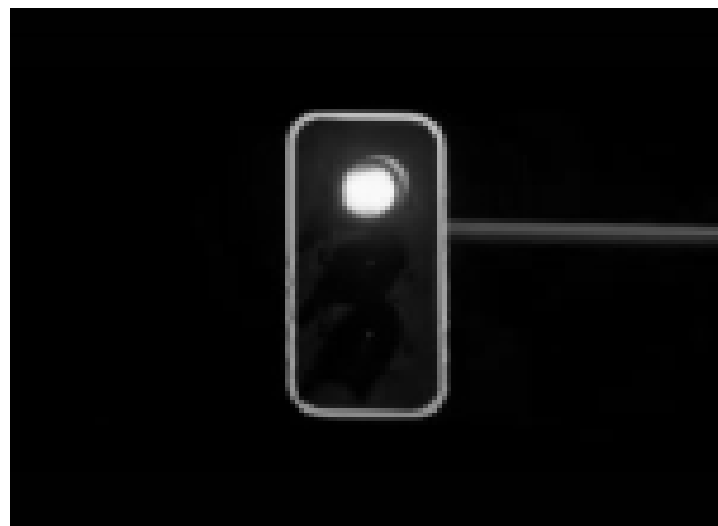
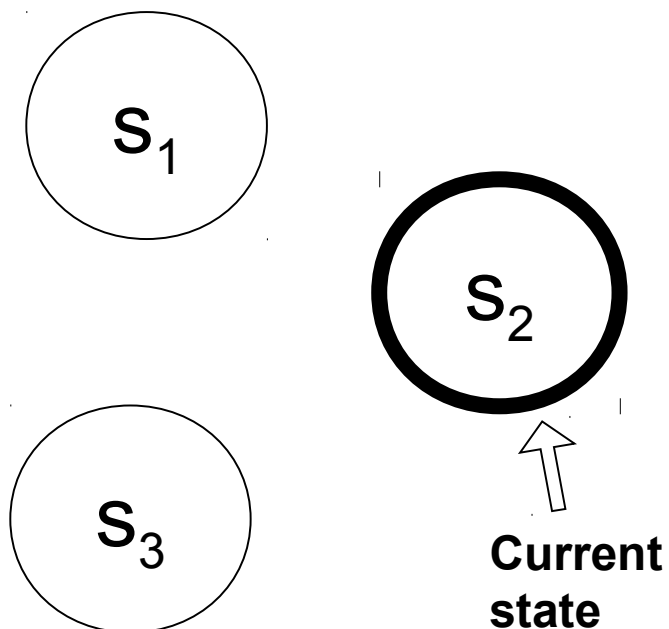
Semaforo: inferenza



$$O_2 = \langle q_2 = s_3, q_1 = s_2 \rangle$$

$$\begin{aligned} \text{Inferenza: } P(O | \lambda) &= P(O) \\ &= P(q_2 = s_3, q_1 = s_2) = P(q_2, q_1) \end{aligned}$$

Semaforo: inferenza



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

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

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

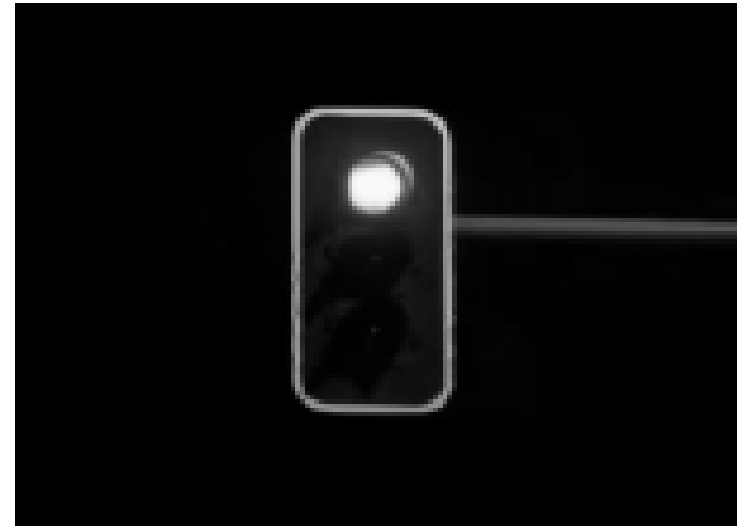
$$= 0.99 * 0.33 = .326$$

Inferenza importante

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

Limiti dei modelli di markov

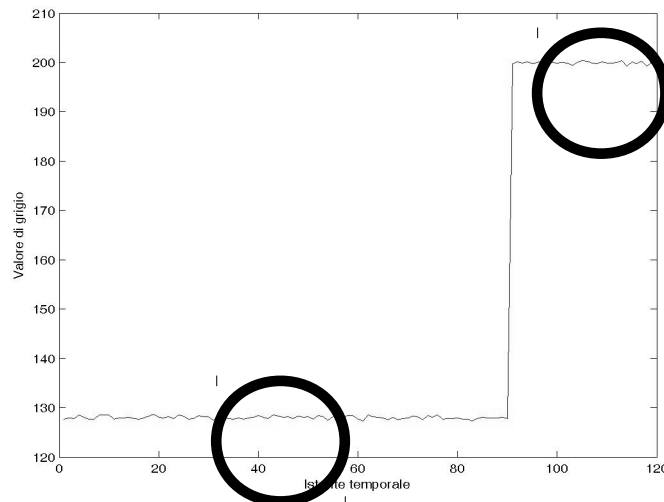
1. Lo stato è sempre osservabile in modo deterministico attraverso le osservazioni (non c'è rumore)



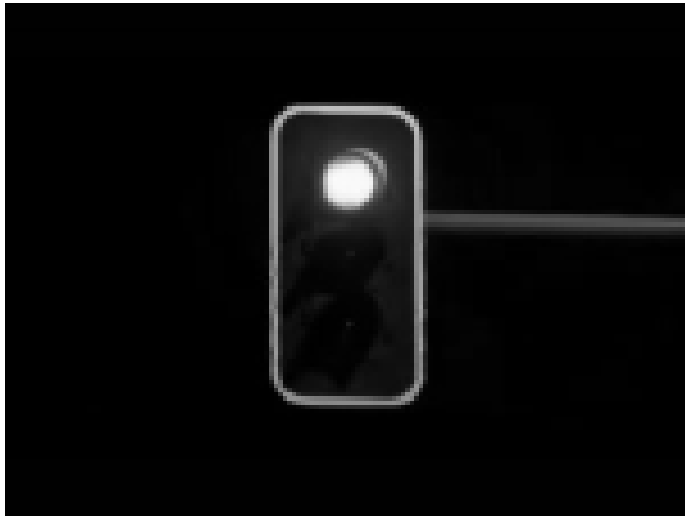
NOTA

STATO = "colore del semaforo"

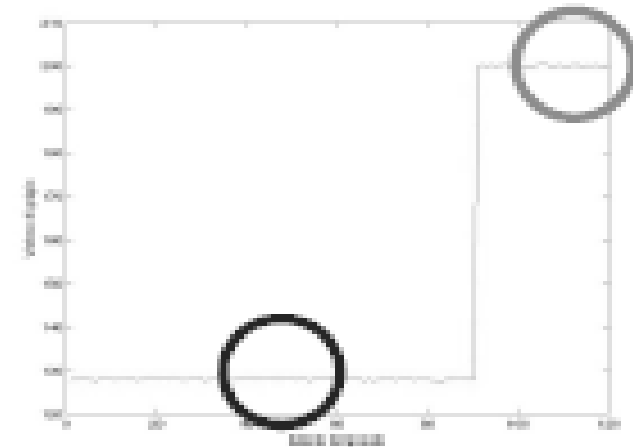
OSSERVAZIONE: "immagine acquisita con la telecamera, da cui posso segmentare la lampada e misurarne il colore"



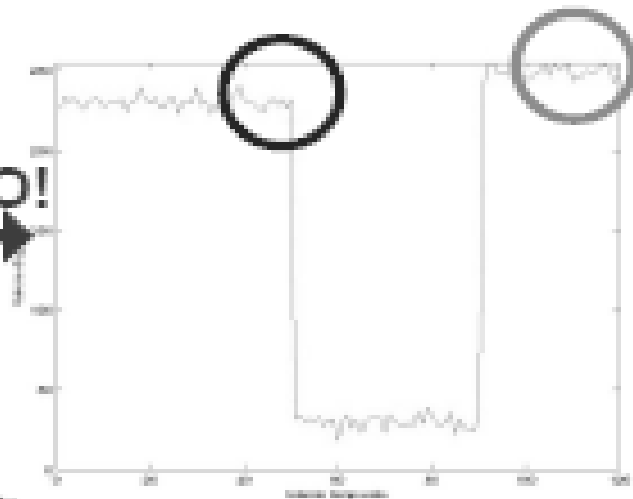
Limiti dei modelli di markov



OK
↓



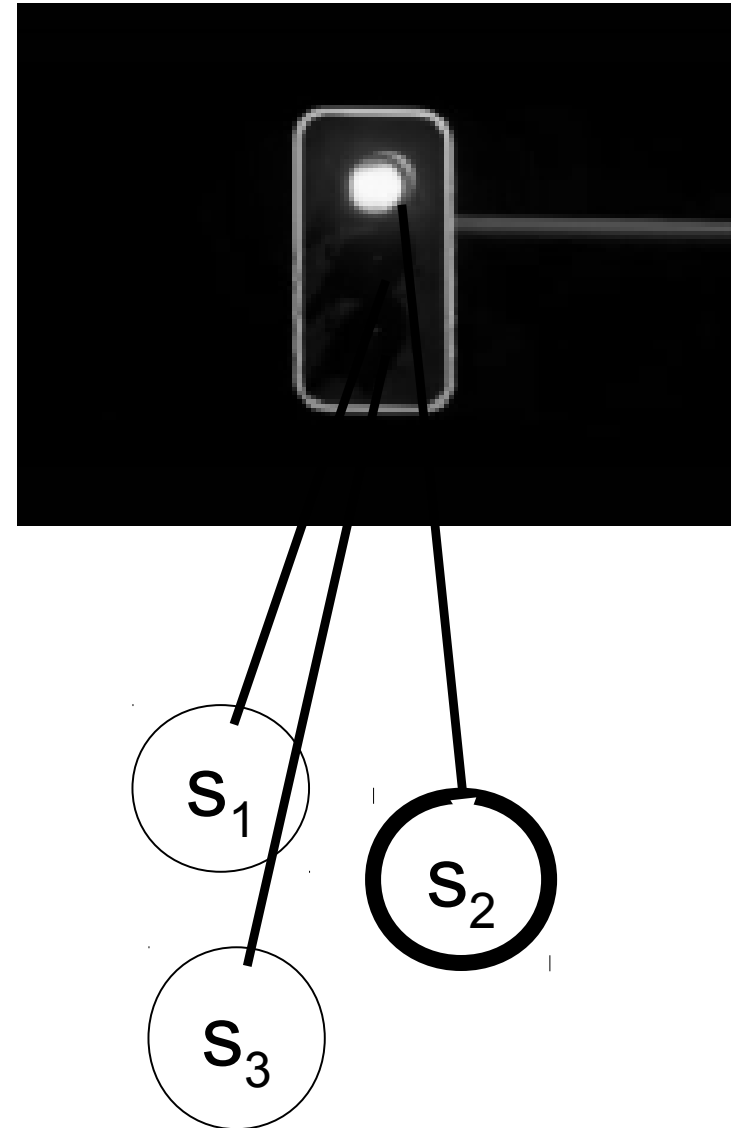
NO!
↓



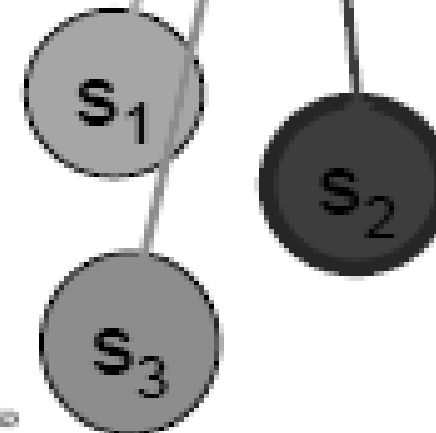
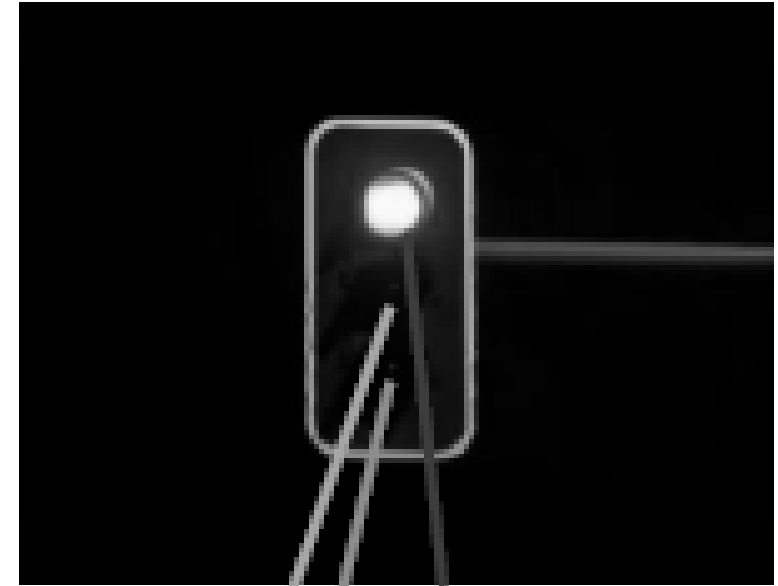
NOISE!

Limiti dei modelli di markov

2. (Più importante!) Nel caso del semaforo, lo stato è sempre esplicito (una particolare configurazione del semaforo) e può essere determinato in maniera univoca dall'osservazione (lo stato corrisponde al colore della lampada!)
- Non sempre vero!



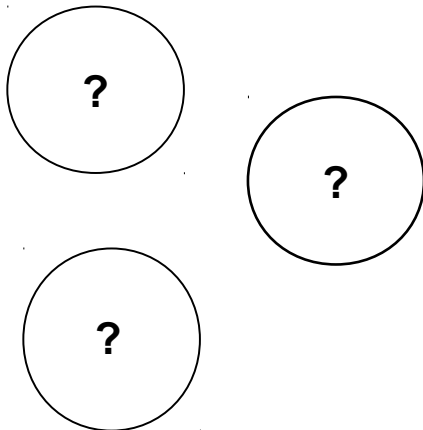
Limiti dei modelli di markov



Dai Modelli di Markov agli HMM



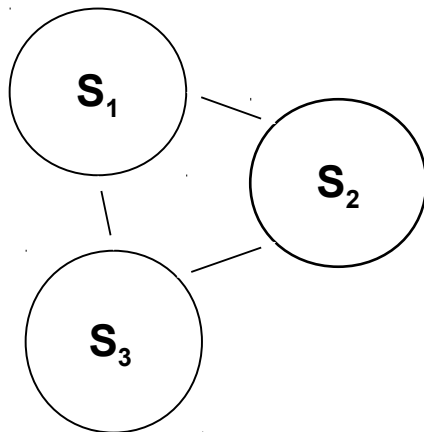
Posso guardare il video di una strada: posso osservare che c'è un sistema che evolve, ma non posso capire qual'è il meccanismo che regola tale evoluzione



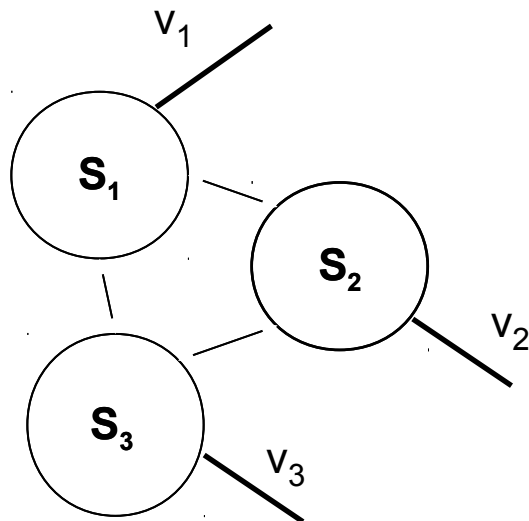
Dai Modelli di Markov agli HMM



Tuttavia riesco ad osservare
che il sistema evolve “a
stati”

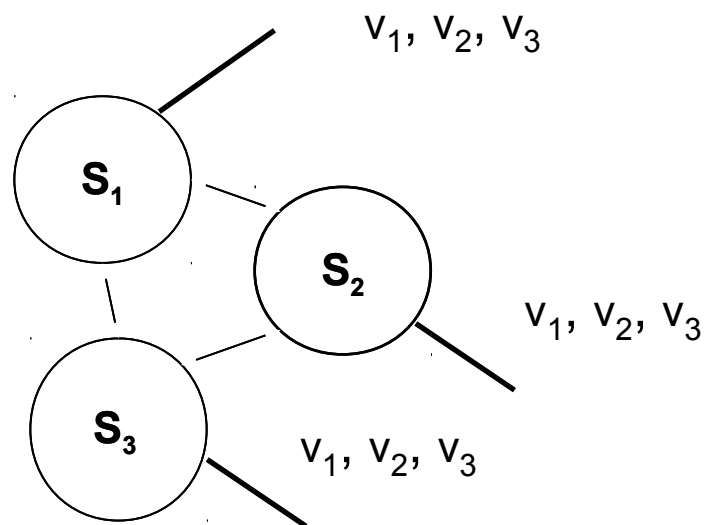


Dai Modelli di Markov agli HMM



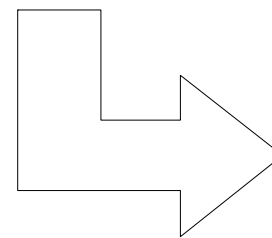
- Meglio: osservo che il sistema evolve grazie ad un insieme di stati “nascosti” (gli stati del semaforo, che io però non posso vedere, sono fuori dal campo di vista della telecamera)
- Posso però osservarne le “conseguenze”, cioè i flussi delle macchine

Dai Modelli di Markov agli HMM



Gli stati sono “entità nascoste” che possono essere identificate solo tramite le osservazioni (il flusso delle macchine)

C'è una relazione tra l'osservazione e lo stato nascosto



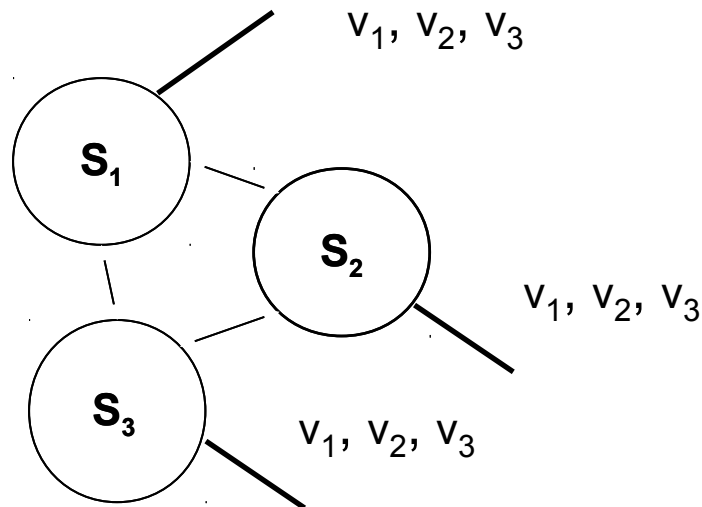
HMM

Dai Modelli di Markov agli HMM



Gli Hidden Markov Models sono modelli che possono modellare in maniera “probabilistica” la dinamica di un sistema senza identificarne le cause

Gli stati sono identificabili solo attraverso le osservazioni, in maniera probabilistica



Processo generativo

Premessa

- ⇒ Processo generativo: come si può usare un modello per generare dei dati
 - ⇒ Esempio: data una gaussiana (un modello per punti) si può generare un punto, basta “campionare” dalla gaussiana
- ⇒ Nel caso di Markov Models e Hidden Markov Models (modelli per dati sequenziali) occorre generare una sequenza

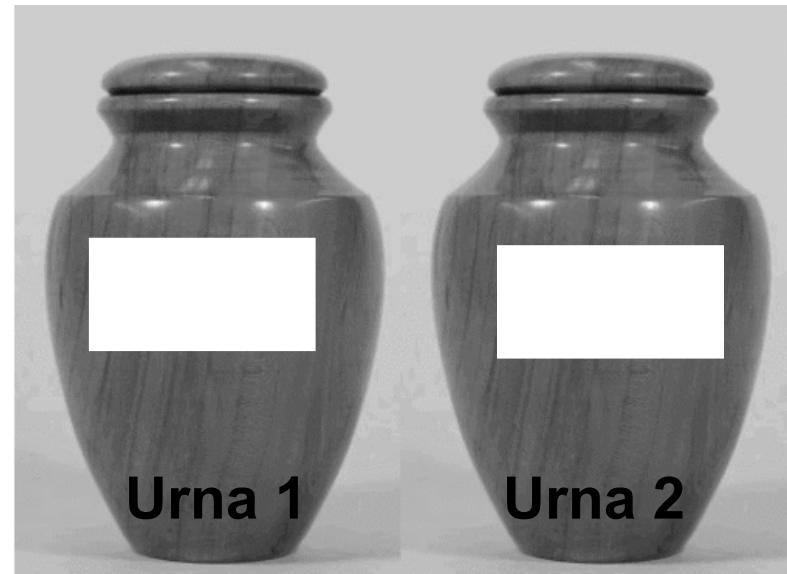
Processo generativo

- ⇒ Si parte da uno stato iniziale
- ⇒ Ad ogni istante:
 - ⇒ Il sistema cambia stato (secondo la matrice di transizione A)
 - ⇒ Viene emesso un simbolo
- ⇒ Con i Markov Models: gli stati sono conosciuti
 - ⇒ Dato uno stato, c'è un solo simbolo che può essere emesso (esempio: stato "giallo" → la camera vedrà il colore giallo)
- ⇒ Con gli Hidden Markov Models: gli stati sono sconosciuti
 - ⇒ Dato uno stato, ogni simbolo può essere emesso con una certa probabilità (esempio: stato "giallo" → la camera può vedere le macchine passare con probabilità 0.95 e le macchine che si fermano con probabilità 0.05)

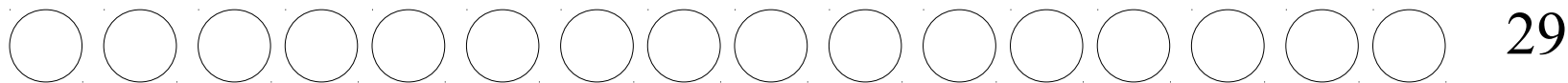
Esempio

- ⇒ Ho due urne che contengono palline colorate:
 - ⇒ Ogni urna rappresenta uno stato
 - ⇒ Le palline sono gli eventi osservabili (ogni urna “emette” palline colorate)
 - ⇒ La matrice di transizione permette di capire come scegliere le urne

Gli stati: le urne

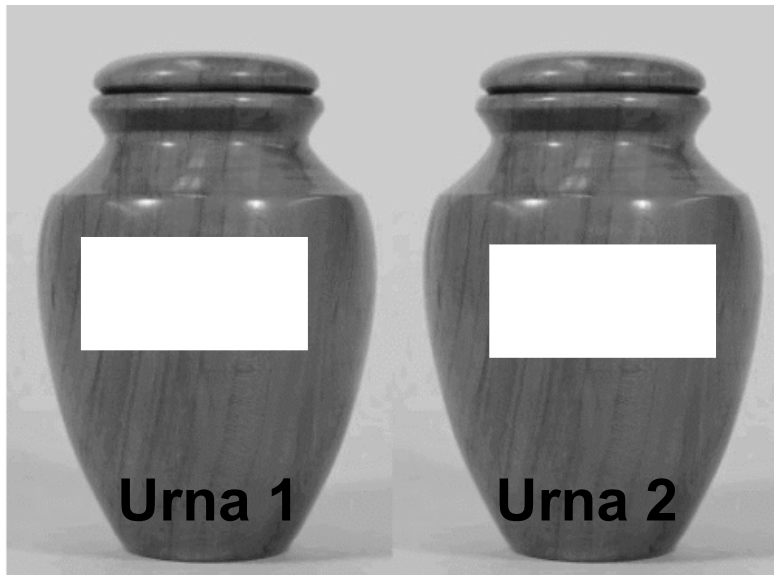


La sequenza di simboli: una
sequenza di palline colorate



Esempio

⇒ Matrice di transizione: spiega come scegliere l'urna al tempo $t+1$ sapendo quale urna è stata scelta al tempo t



0.7	0.3
0.1	0.9

a_{11} : Probabilità di scegliere la prima urna al tempo t sapendo che al tempo $t-1$ ho scelto la prima urna

a_{12} : Probabilità di scegliere la seconda urna al tempo t sapendo che al tempo $t-1$ ho scelto la prima urna

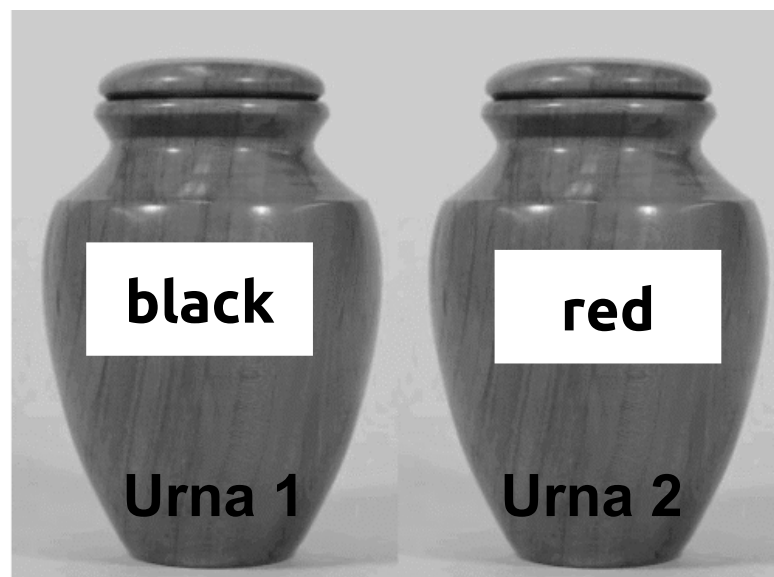
Esempio

Processo generativo per generare una sequenza di T palline:

- ⇒ Scegliere un'urna utilizzando la “probabilità iniziale”
- ⇒ Estrarre una pallina da quella urna (con re-inserimento): rappresenta la prima pallina della sequenza
- ⇒ Ripetere per $T-1$ passi
 - ⇒ Cambiare urna utilizzando la probabilità di transizione
 - ⇒ Estrarre una pallina dall'urna scelta (con re-inserimento) e aggiungerla alla sequenza

Con i modelli di Markov

⇒ Ogni urna contiene solo palline di un singolo colore



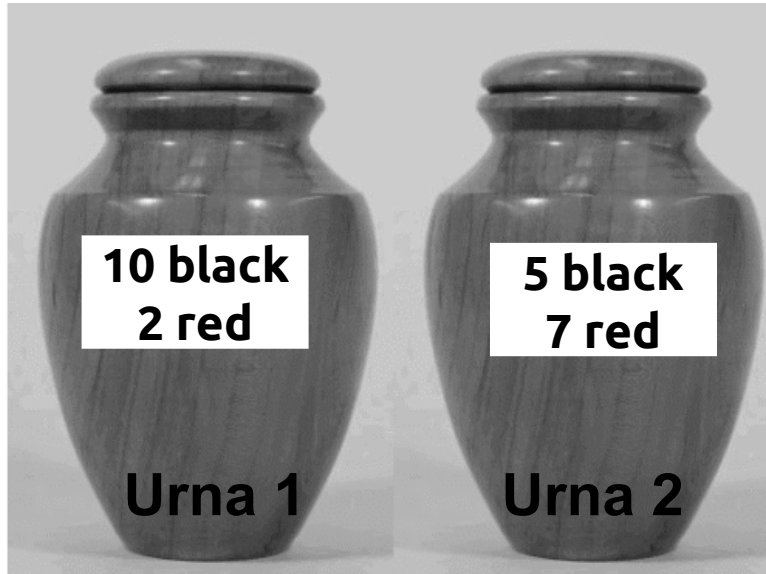
⇒ Se viene scelta la prima urna verrà estratta sicuramente una pallina nera

⇒ CONSEGUEZZA: Se si osserva una sequenza di palline (simboli) si può derivare perfettamente la sequenza di urne (stati) → **gli stati sono osservabili**



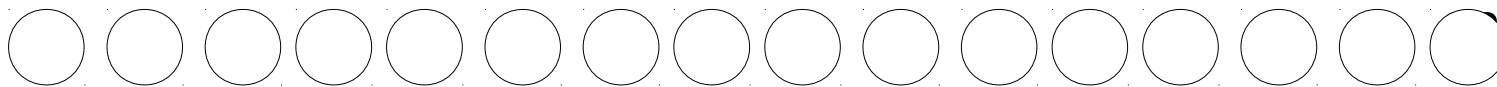
Con gli HMM

⇒ Ogni urna contiene palline di colori differenti

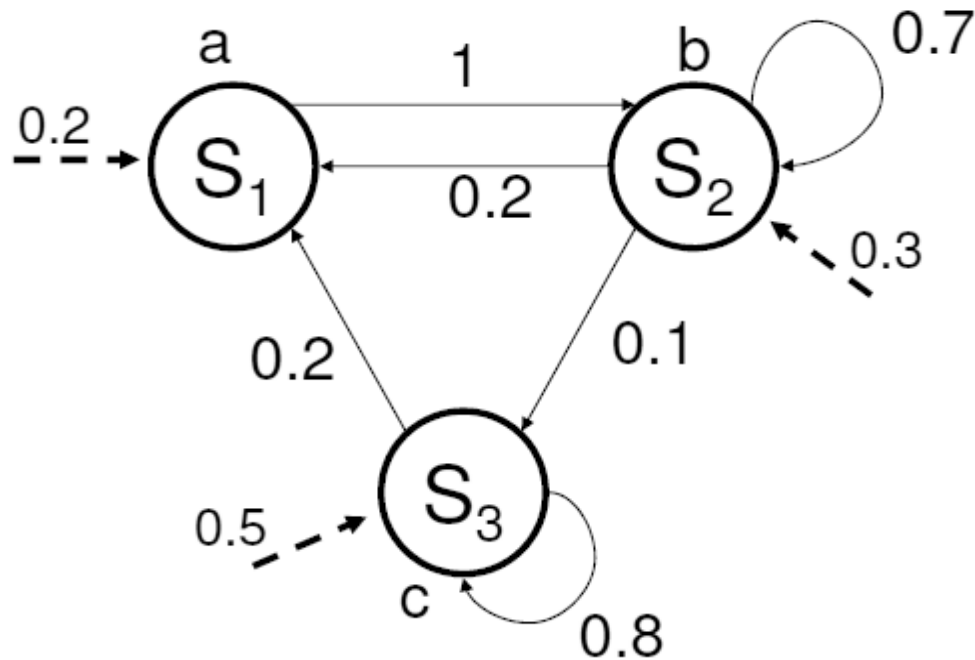


⇒ Se viene scelta la prima urna, verrà estratta una pallina nera con probabilità $10/12$ e una pallina rossa con probabilità $2/12$

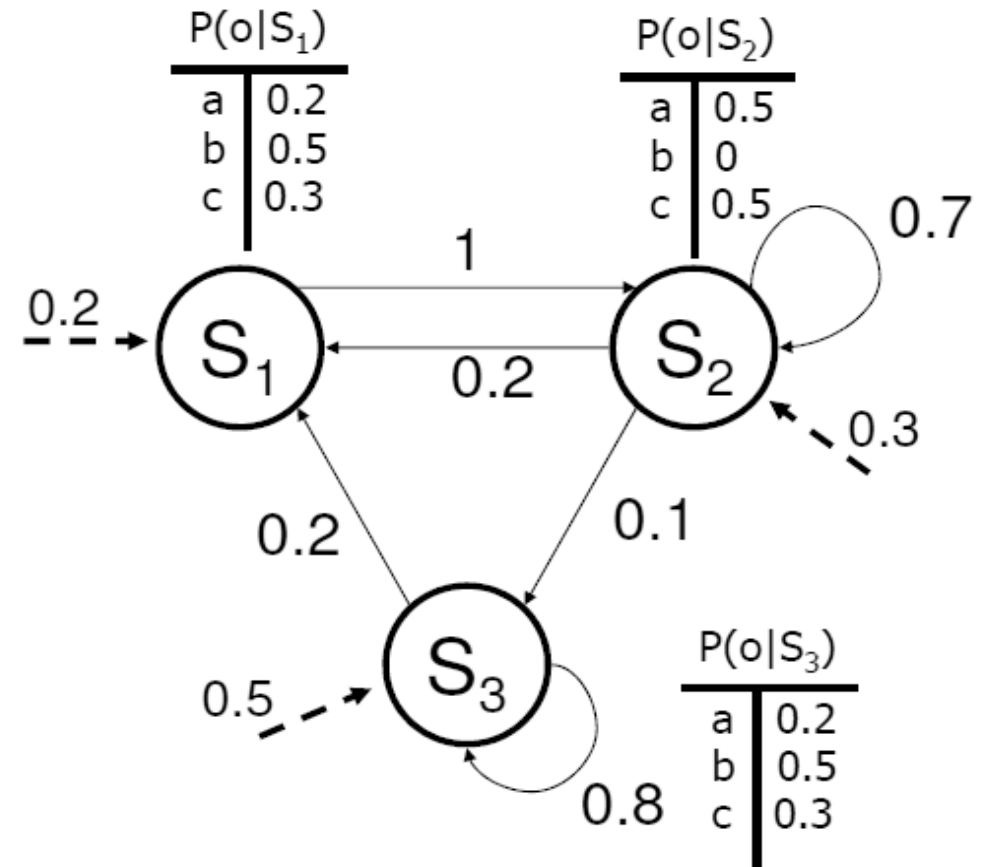
⇒ CONSEQUENZA: Se si osserva una sequenza di palline (simboli) non si può inferire con certezza la sequenza degli stati (le urne): **gli stati sono nascosti!**



Riassumendo



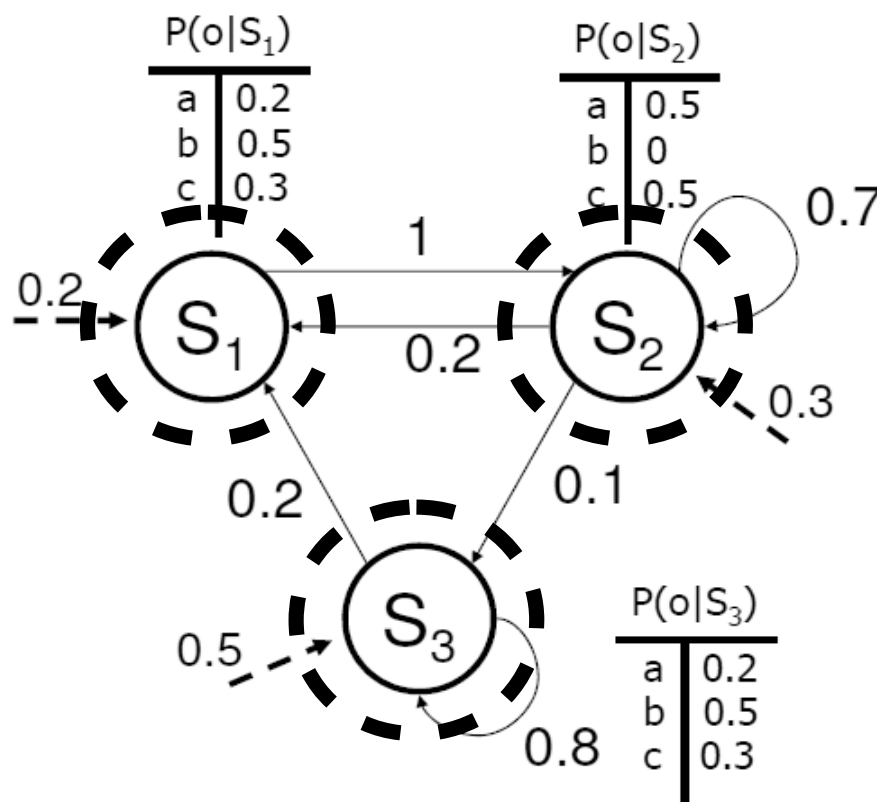
Markov Models



Hidden Markov Models

Componenti di un HMM

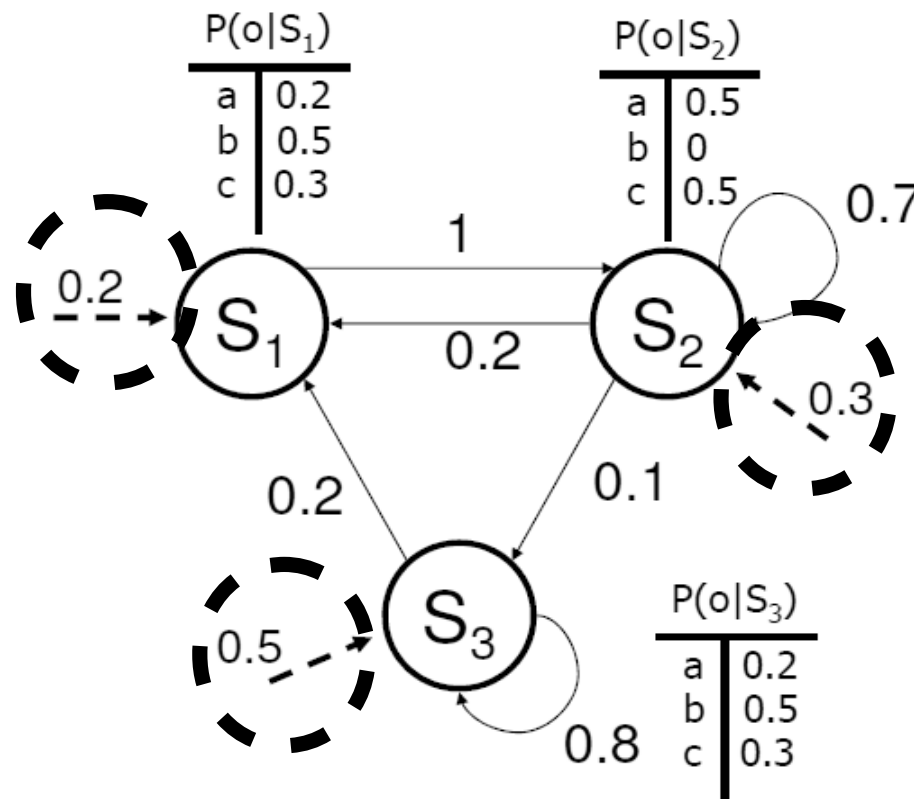
- ⇒ Un insieme $S = \{S_1, S_2, \dots, S_N\}$ di stati (nascosti)
- ⇒ Anche se nascosti, per molte applicazioni pratiche si può inferire un significato fisico



Componenti di un HMM

⇒ Una probabilità degli stati iniziali $\pi = \{\pi_i\}$

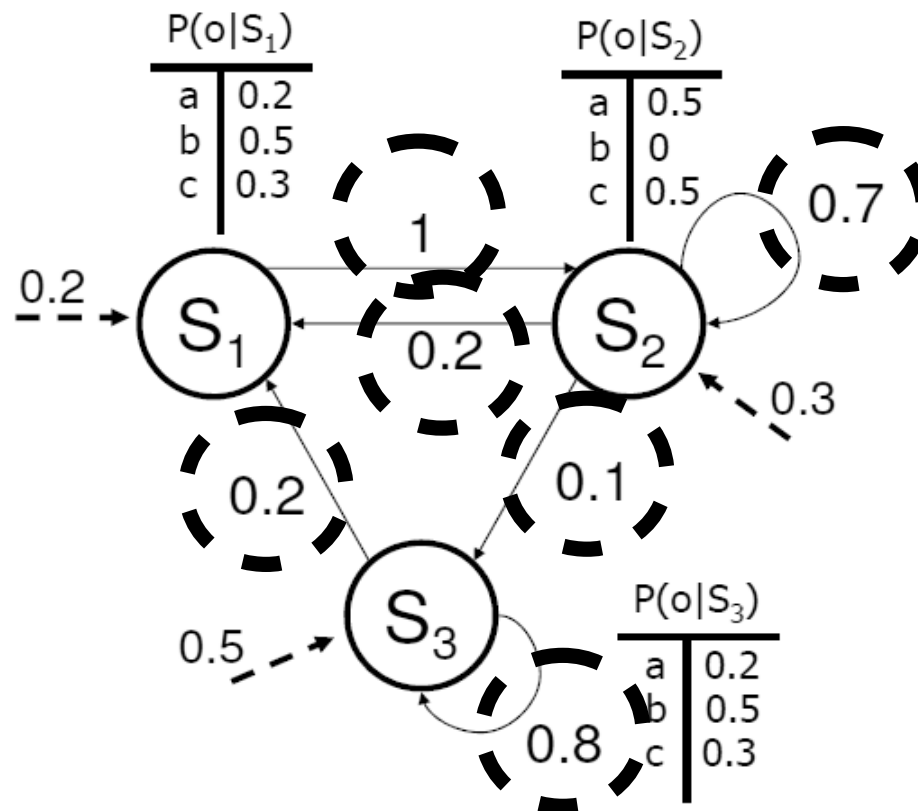
$$\pi_i = P(Q_1=S_i) \quad 1 \leq i \leq N$$



Componenti di un HMM

⇒ Una probabilità di transizione tra gli stati $\mathbf{A} = \{a_{ij}\}$

$$a_{ij} = P(Q_t = S_j | Q_{t-1} = S_i) \quad 1 \leq i, j \leq N$$



Componenti di un HMM

⇒ Una probabilità di emissione dei simboli $\mathbf{B} = \{b_j(v)\}$

$$b_j(v) = P(v \text{ è emesso al tempo } t \mid Q_t = S_j) \\ v \in V$$

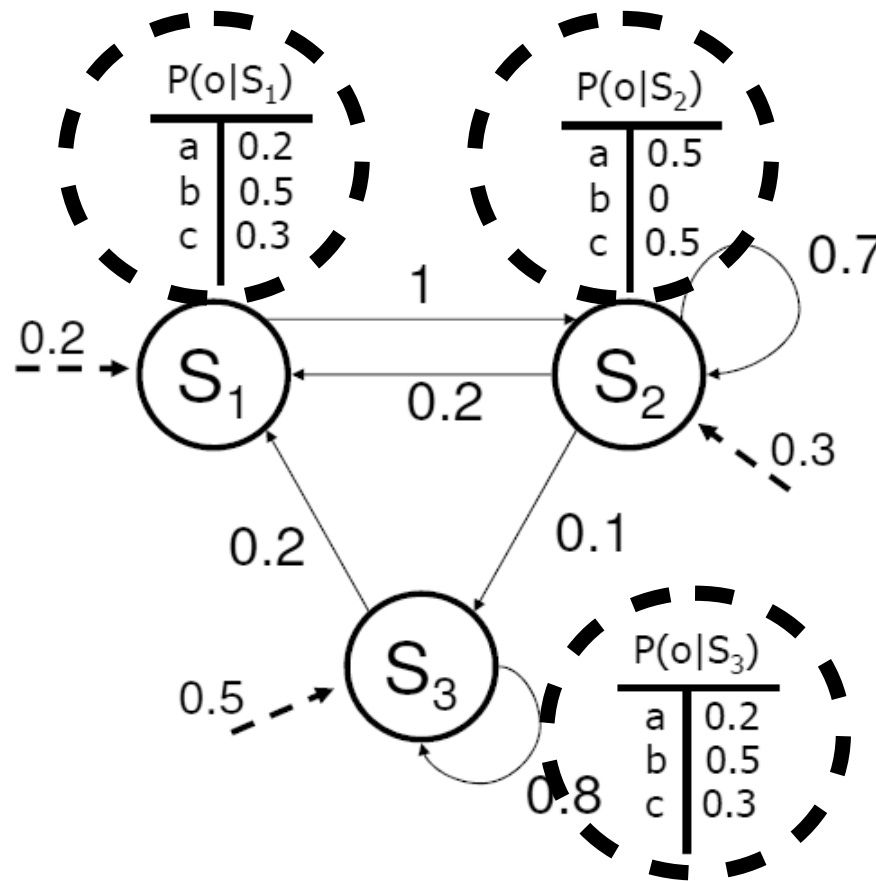
⇒ $\mathbf{V}$ è l'insieme dei simboli, che rappresenta l'output fisico osservabile del sistema

⇒ Caso discreto, e.g. $\mathbf{V} = \{v_1, \dots, v_M\} \rightarrow \text{HMM DISCRETO}$

⇒ Continuo, e.g. $\mathbf{V} = \mathbb{R} \rightarrow \text{HMM CONTINUO}$

⇒ $b_j(v)$ può essere una gaussiana ($\mathbf{V}$ continuo)

Componenti di un HMM



Problemi di base degli HMM

- ⇒ *Valutazione*: data una sequenza $\mathbf{O}$ e un modello λ , calcolare $P(\mathbf{O}|\lambda)$
 - ⇒ Soluzione: Forward-Backward procedure

- ⇒ *Decodifica*: data una sequenza $\mathbf{O}$ e un modello λ , calcolare la sequenza “ottimale” di stati $Q_1, \dots, Q_T$ che ha generato la sequenza $\mathbf{O}$
 - ⇒ Soluzione: Algoritmo di Viterbi

- ⇒ *Addestramento*: dato un insieme di sequenze $\{\mathbf{O}_i\}$, determinare il miglior modello λ , i.e. il modello per cui $P(\{\mathbf{O}_i\}|\lambda)$ è massimizzata
 - ⇒ Soluzione: Baum-Welch (EM)

Riassumendo: HMM per la classificazione

⇒ Dato un problema a C classi:

⇒ Training:

- utilizzare tutte le sequenze della classe C_k per addestrare l'HMM che rappresenta quella classe

⇒ Testing:

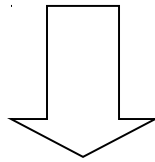
- dato un oggetto sconosciuto, si classifica con la regola di Bayes
 - si calcola la likelihood dell'oggetto con ogni HMM (eventualmente si moltiplica per il prior)
 - si assegna l'oggetto alla classe la cui posterior è massima

Questioni avanzate:

1. Inizializzazione

Baum-Welch implementa una discesa lungo il gradiente (ottimizzatore locale)

⇒ La log likelihood è altamente multimodale (con molti massimi)



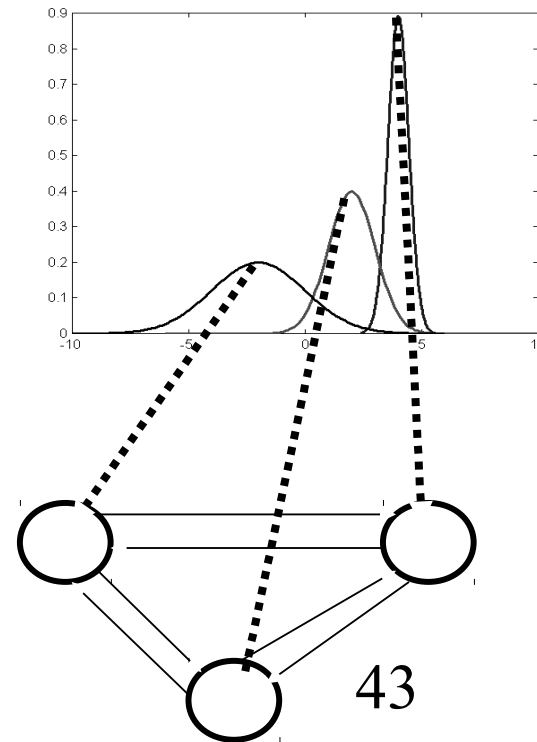
⇒ L'inizializzazione dei parametri può essere cruciale!

Questioni avanzate:

1. Inizializzazione

- HMM discreto: diverse inizializzazioni casuali + prendere il migliore
- HMM continuo: problema tosto!

Soluzione: clustering dei simboli con GMM e inizializzazione con i risultati del clustering



GMM
clustering

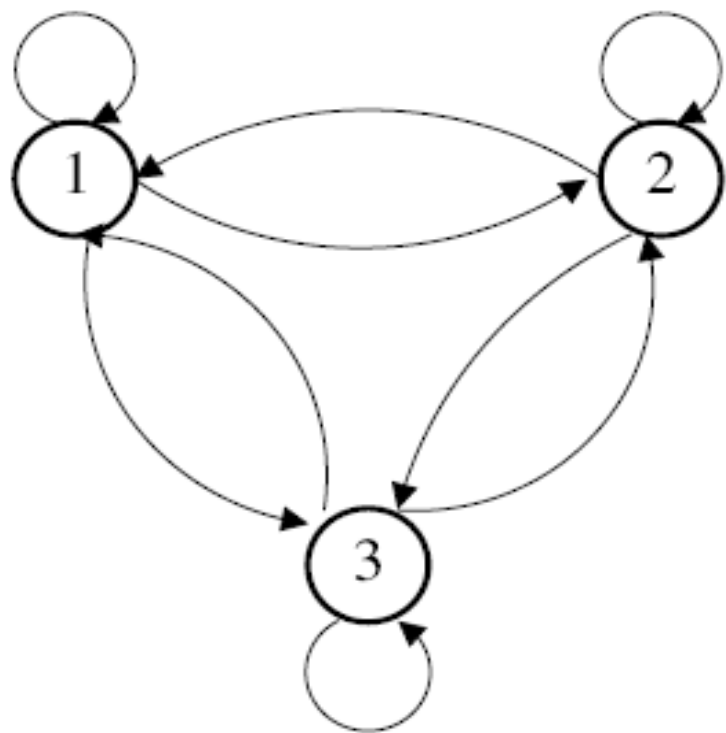
Questioni avanzate:

2. Model Selection

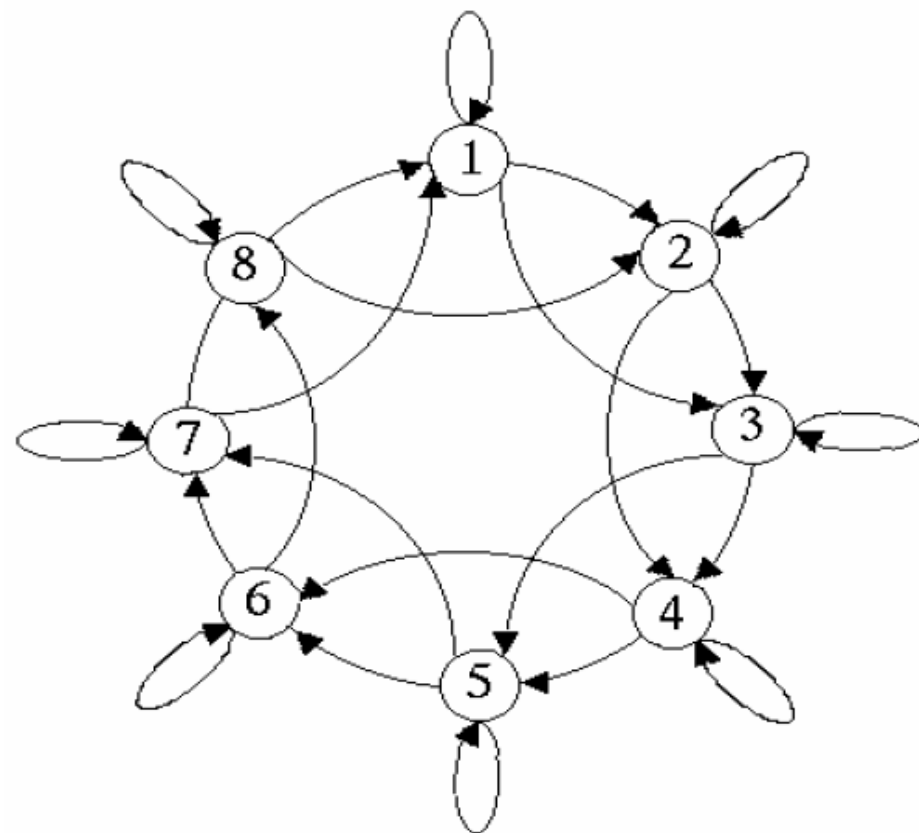
Il problema di determinare la struttura dell'HMM

Scegliere il numero di stati: il problema “standard” di model selection (per prevenire l'overtraining)

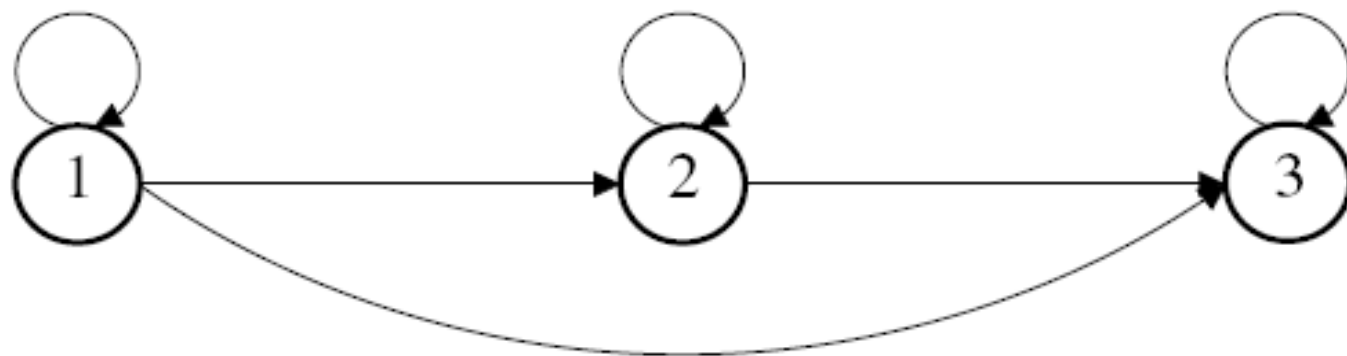
Scegliere la topologia: forzare la presenza o l'assenza di connessioni particolari



standard ergodic
HMM



circular HMM [Arica,Yarman-Vural
ICPR00]



Left to right HMM
[Jelinek, Proc. IEEE 1976]

HMM in bioinformatics: Pair hidden Markov models

Problema dell'allineamento di sequenze

- **INPUT:** Due sequenze su di un alfabeto Σ (4 nucleotidi or 20 aminoacidi):

ATGTTAT e ATCGTAC

- **OUTPUT:** Inserendo '-' e shiftando le due sequenze devo produrre due sequenze allineate della stessa lunghezza, massimizzando il punteggio di allineamento

```
A T - G T T A T
A T C G T - A C
```

Scoring Matrix

	A	R	N	K
A	5	-2	-1	-1
R	-	7	-1	3
N	-	-	7	0
K	-	-	-	6

AKRANR

KAAANK

$$-1 + (-1) + (-2) + 5 + 7 + 3 = 11$$

- Come si calcola il punteggio di allineamento: i match valgono molto, i mismatch e i gap poco (o peso negativo)
- I pesi dipendono dal particolare aminoacido (nucleotide)
- Si tiene conto dell'evidenza biologica

Altre penalizzazioni...

- In natura spesso gli indels ('-') appaiono raggruppati (affine indels)
- Posso penalizzare di più allineamenti in cui ho tanti singoli indels

ATA- -GC
ATATTGC

ATAG -GC
AT- GTGC

**Utilizzando un normale punteggio questi 2
allineamenti sono equiprobabili**

Allineamento di sequenze tramite HMM

- Le PAIR-HMM risolvono questo compito incorporando tutte le idee viste in precedenza
 - Penalizzazioni per mismatch e indels – Affine indels
- Principale differenza tra una Pair HMM e un HMM.
 - HMM modella una sequenza mentre PAIR HMM modella una coppia di sequenze: il simbolo di osservazione è una coppia (due simboli oppure un simbolo e un gap)
 - Il numero di stati è fissato a priori: 3 (uno stato per la situazione in cui si ha un match, uno per quando c'è un gap nella prima sequenza, uno per quando c'è un gap nella seconda)

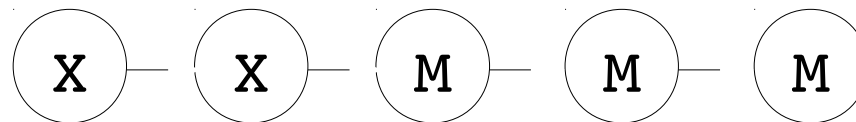
Componenti di una pair HMM

Stati nascosti

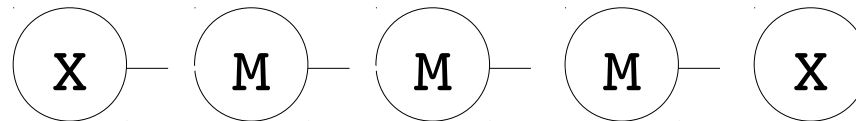
- Match (M): le due sequenze hanno un match
- Insertion in x (X): c'è un gap nella seconda sequenza
- Insertion in y (Y): c'è un gap nella prima sequenza

NOTA: uno specifico allineamento corrisponde ad una ben precisa sequenza di stati di una pair HMM

T	T	C	C	G
-	-	C	C	C



T	T	C	C	G
-	C	C	C	-



Componenti di una pair HMM

Simboli osservati

Ogni “observation symbol” dell’HMM è una coppia ordinata di 2 lettere o di una lettera e un gap

- Match (M): $\{(a,b) \mid a,b \in \Sigma\}$.
- Insertion in x (X): $\{(a,-) \mid a \in \Sigma\}$.
- Insertion in y (Y): $\{(-,a) \mid a \in \Sigma\}$.

Probabilità di emissione

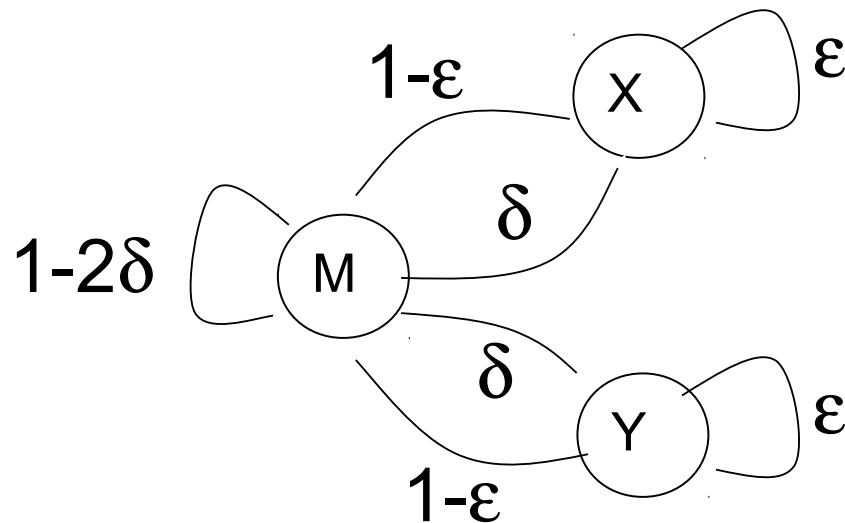
- i. Match: (a,b) con p_{ab} – solo per lo stato M
- ii. Insertion in x : $(a,-)$ con q_a – solo per lo stato X
- iii. Insertion in y : $(-,a)$.con q_a - solo per lo stato Y

Componenti di una pair HMM

Probabilità di transizione

δ = probabilità del 1st '-'

ε = probabilità di estendere '-'



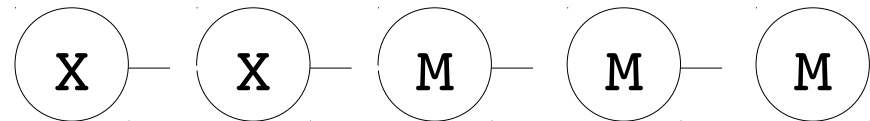
	M	X	Y
M	$1-2\delta$	δ	δ
X	$1-\varepsilon$	ε	0
Y	$1-\varepsilon$	0	ε

Punteggio di un allineamento utilizzando una pair-HMM

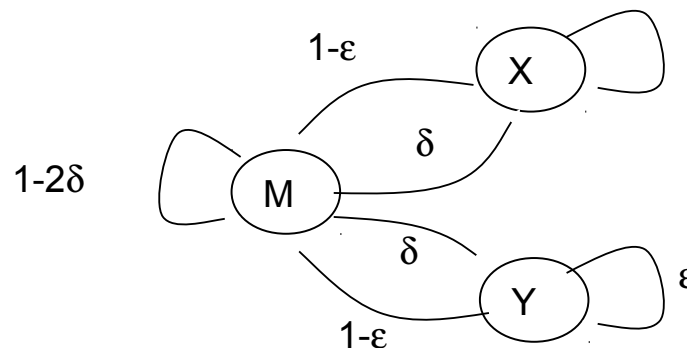
RICAPITOLANDO:

Uno specifico allineamento corrisponde ad una ben precisa sequenza di stati di un pair HMM

T	T	C	C	G
-	-	C	C	C



Ad ogni allineamento può essere assegnato un punteggio: la probabilità di osservare quella sequenza dato il modello



Punteggio di un allineamento utilizzando una pair-HMM

Date due sequenze si può utilizzare la pair HMM per cercare l'allineamento più probabile (quello che massimizza la probabilità di allineamento)

In altre parole: ci interessa la sequenza di stati ottimale (algoritmo di Viterbi!)

NOTA: I parametri della pair HMM devono rappresentare un sistema di punteggio che rifletta un modello evolutivo, o quantomeno “biologicamente” sensato

HMM in bioinformatics:

Profile HMM

Introduzione

- Idea: costruire un'architettura in grado di modellare un insieme di sequenze (tipicamente proteiche), con inserzioni e delezioni
- L'idea principale è che si vuole modellare le inserzioni e le delezioni in maniera dipendente dalle posizioni (cosa che il Pair HMM non riesce a fare)
- Un profile HMM ha una topologia strettamente “left to right”
 - Per ogni sito, si hanno tre possibili stati: uno di Match, uno di inserzione e uno di cancellazione

Profile HMM

- IDEE principale:
 - Si parte da un allineamento multiplo di un insieme di sequenze correlate
 - Si decide qual'è la lunghezza “corretta” (attesa) di una sequenza in quella famiglia
 - Si costruisce un HMM con un numero di stati di match pari alla lunghezza corretta (questo modello si chiama “ungapped” HMM)
 - Si gestiscono inserzioni e delezioni aggiugnendo stati al modello

Profile HMM

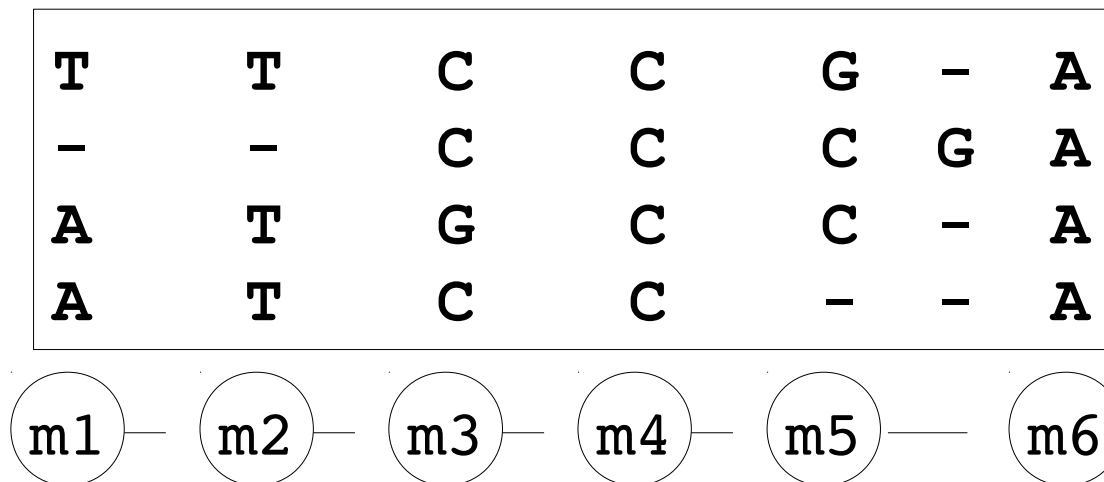
- Lunghezza “corretta” (attesa) di una sequenza nella famiglia delle sequenze allineate
 - Ad esempio: si guardano le colonne che hanno un numero “sufficientemente alto” di caratteri allineati

T	T	C	C	G	-	A
-	-	C	C	C	G	A
A	T	G	C	C	-	A
A	T	C	C	-	-	A



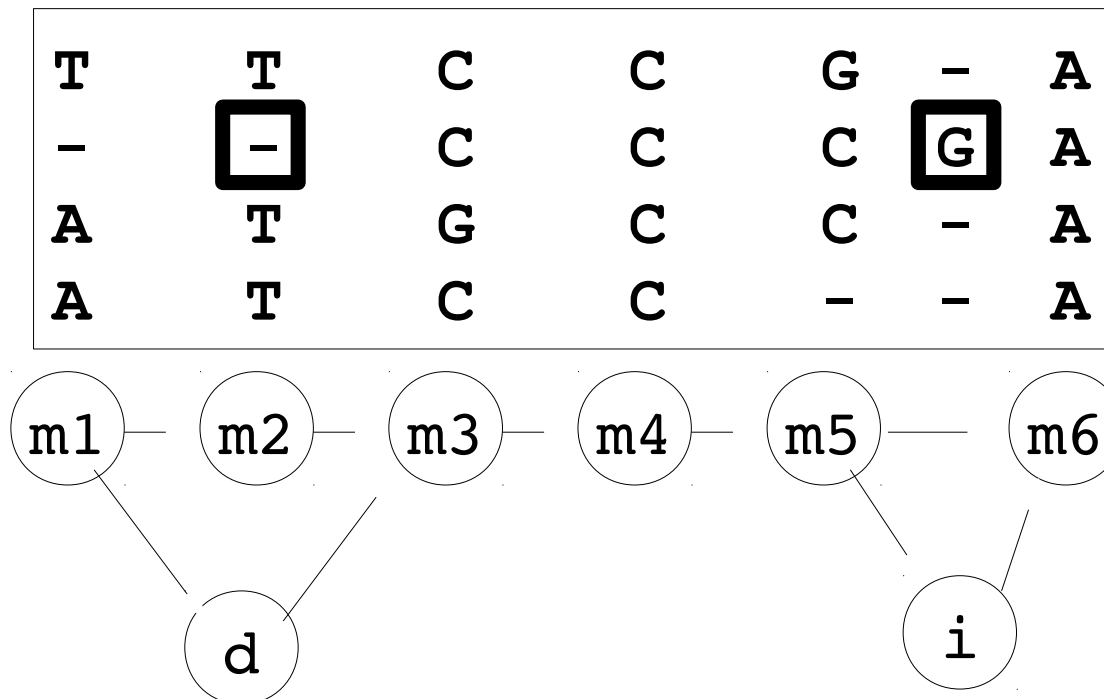
Profile HMM

- Si costruisce un HMM con un numero di stati di Match pari alla lunghezza stimata



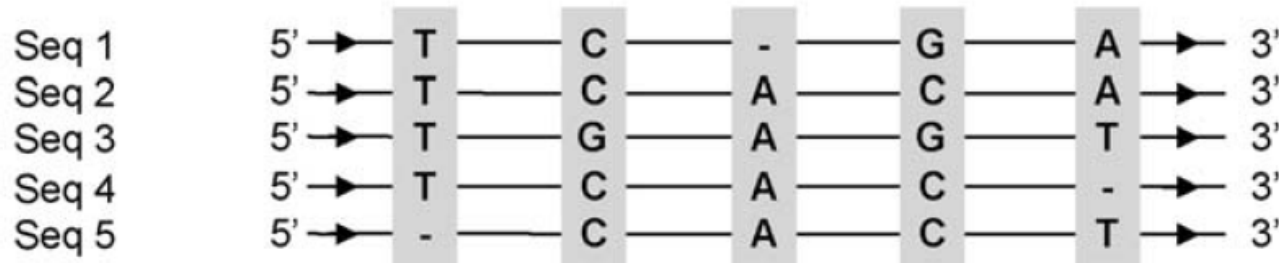
Profile HMM

- Le inserzioni o le cancellazioni vengono poi gestite con stati apposta
 - Esempio:
 - L' inserzione di G è uno stato tra m5 e m6
 - La cancellazione è uno stato alternativo a m2

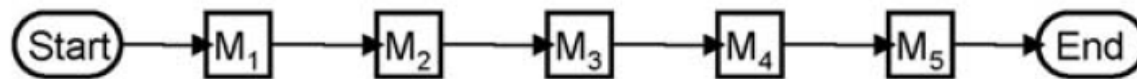


Profile HMM

(a) Sequence Alignment

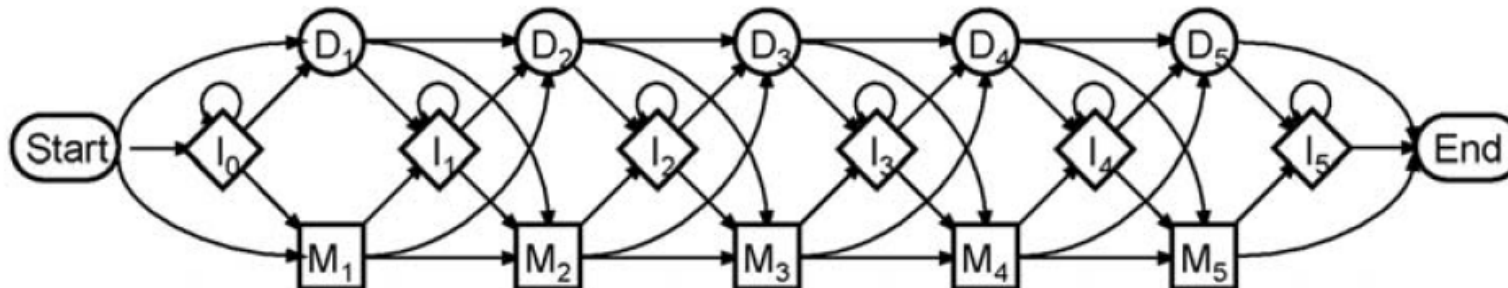


(b) Ungapped HMM



M_k **Match states**

(c) Profile-HMM



M_k **Match states**

I_k **Insert states**

D_k **Delete states**

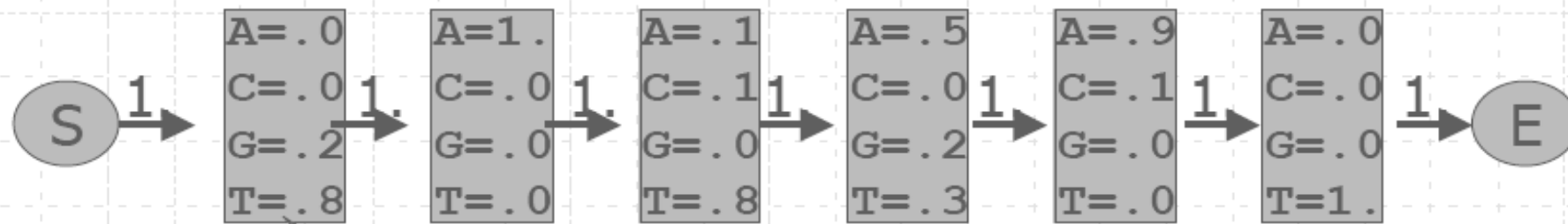
Profile HMM

Dato un allineamento multiplo, come si stimano le probabilità del profile HMM?

⇒ Occorre stimare:

- ⇒ Le probabilità di emissione di ogni simbolo in ogni stato
- ⇒ Le probabilità di transizione tra gli stati

vediamo la stima delle probabilità di emissione nel caso senza gaps (quindi senza gli stati di insertion e deletion)



A State per Site

A	0	10	1	5	9	0
C	0	0	1	0	1	0
G	2	0	0	2	0	0
T	8	0	8	3	0	10

T	A	T	G	A	T
T	A	T	A	A	T
T	A	T	A	A	T
T	A	T	A	A	T
T	A	T	T	A	T
G	A	T	A	A	T
G	A	T	A	C	T
T	A	C	G	A	T
T	A	T	T	A	T

tyr tRNA
 rrn D1
 rrn X1
 rrn (DXE)₂
 rrn C1
 rrn A1
 rrn A2
 λ Pr
 λ PL
 T7 A3
 T7 A1
 T7 A2
 fd VIII

TCTCAACGTAACAC TTTACAGCGGCG • • CGTCATTTGATATGATGC • GCCCCGCTTCCCGATAAGGG
 GATCAAAAAAATAC TTGTGCAAAAAA • • TTGGGATCCCTATAATGCGCCTCCGT TGAGACGACAACG
 ATGCATTTTTTCCGC TTGTCTTTCCTGA • • GCCGACTCCCTATAATGCGCCTCCATCGACACGGCGGAT
 CCTGAAATTCAGGG TTGACTCTGAAA • • GAGGAAAGCGTAATATAC • GCCACCTCGCGACAGTGAGC
 CTGCAATTTTTCTATTGCGGCCTGCC • • CAGAACTCCCTATAATGCCCTCCATCGACACGGCGGAT
 TTTTAAATTTCTCTTTGTCAGGCGCG • • AATAACTCCCTATAATGCGCCACCACTGACACGGAACAA
 GCAAAAATAAATGCTTGACTCTGTAG • • CGGGAAGGCGTATTATGC • ACACCCCGCGCGCTGAGAA
 TAACACCGTGCGTGTGTTGACTATTTTA • CCTCTGGCGGTGATAATGG • • TTGCATGTACTAAGGAGGT
 TATCTCTGGCGGTGTTGACATAAATA • CCACTGGCGGTGATACTGA • • GCACATCAGCAGGACGCAC
 GTGAAACAAAACGGTTGACAACATGA • AGTAAACACGGTACGATGT • ACCACATGAAACGACAGTGA
 TATCAAAAAGAGTATTGACTTAAAGT • CTAACCTATAGGATACTTA • CAGCCATCGAGAGGGACACG
 ACGAAAAACAGGTATTGACAACATGAAGTAACATGCAGTAAGATAC • AAATCGCTAGGTAACTAG
 GATACAAATCTCCGTTGTACTTTGTT • • TCGCGCTTGGTATAATCG • CTGGGGGTCAAAGATGAGTG

-35

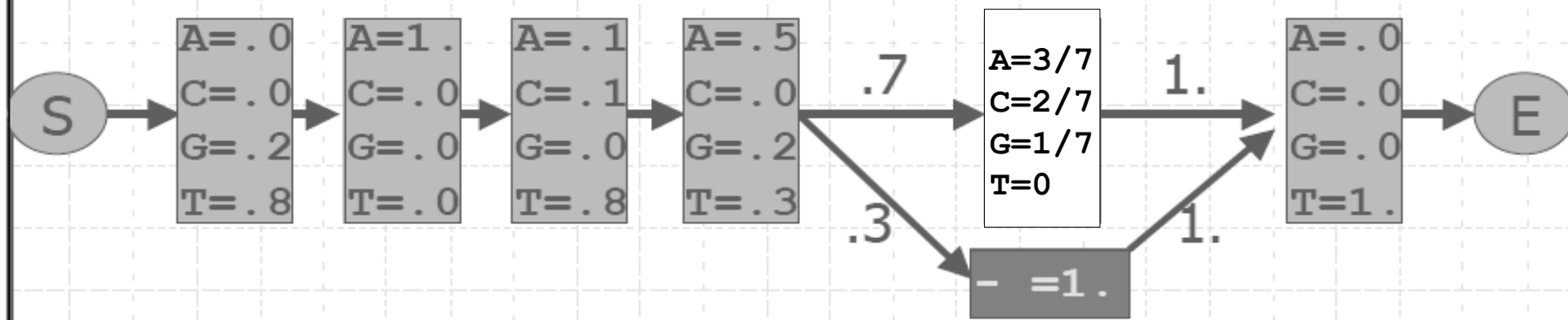
-10

+1

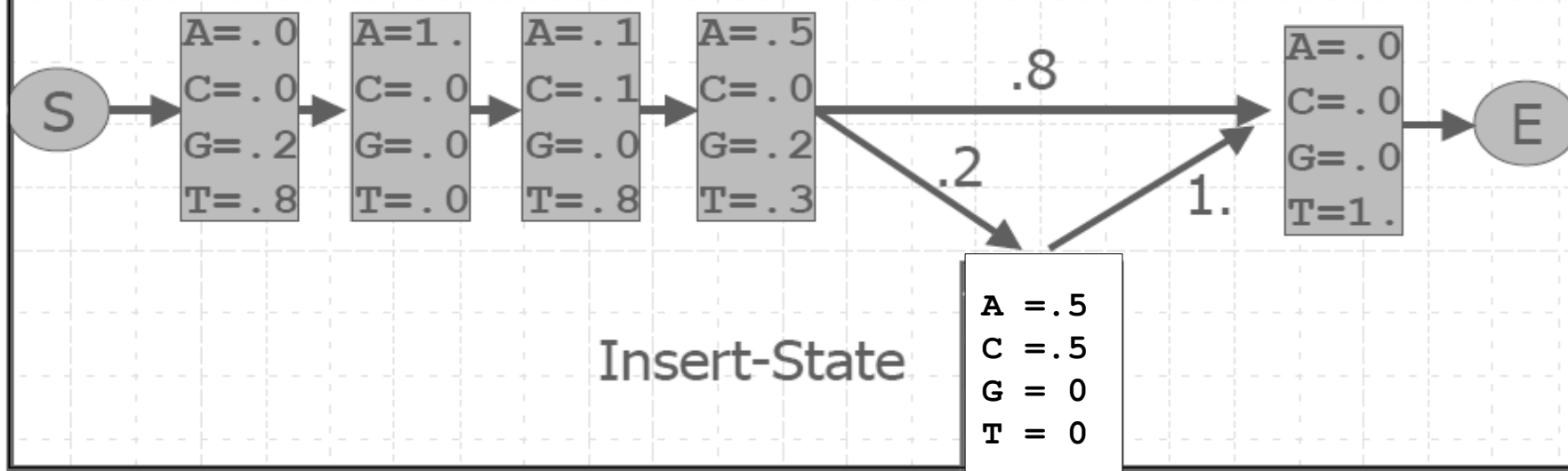
U4

Caso con GAPS

- Deletion = hidden delete state
- Insertion = hidden insert state



T	A	T	G	A	T
T	A	T	A	C	T
T	A	T	A	A	T
T	A	T	T	A	T
T	A	T	A	G	T
T	A	T	T	-	T
G	A	T	A	A	T
G	A	T	A	-	T
T	A	C	G	-	T
T	A	T	T	C	T



T	A	T	G	A	T
T	A	T	A	C	T
T	A	T	A	-	T
T	A	T	T	-	T
T	A	T	T	-	T
T	A	T	T	-	T
G	A	T	A	-	T
G	A	T	A	-	T
T	A	C	G	-	T
T	A	T	T	-	T

Riassumendo

- Una profile HMM rappresenta un HMM che contiene tre tipologie di stati:
 - Match, Insertion and Delete
- L'alfabeto della HMM consiste nei 20 simboli degli aminoacidi e nel simbolo delete (“-”)
- Gli stati “delete” emettono solamente “-” con probabilità 1
- Ogni stato “insert” o “match” emette uno dei 20 aminoacidi ma non il simbolo “-”.

Utilizzi

- Caratterizzazione di una famiglia di sequenze
- Allineamento tra due sequenze (si calcola il percorso di viterbi e si allineano le sequenze di stati di match)
- Confronto tra diverse famiglie (confronto di profili)
- ...