

Riconoscimento e Recupero dell'Informazione per Bioinformatica

LAB. 10 – Hidden Markov Model

Pietro Lovato

Corso di Laurea in Bioinformatica
Dip. di Informatica – Università di Verona
A.A. 2016/2017

Esercizio

Immaginiamo il seguente scenario:

- Gli stati (nascosti) sono le lampade di un semaforo
- Le osservazioni sono i flussi di macchine (ci sono macchine ferme o macchine in movimento)

Stati

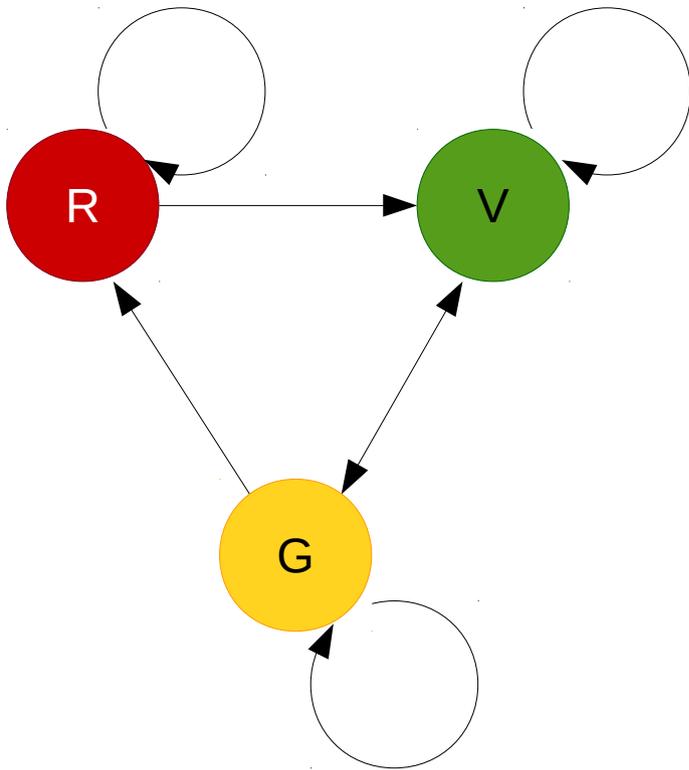


Osservazioni



Esercizio

Il seguente scenario potrebbe essere descritto dal seguente modello di Markov a stati nascosti:

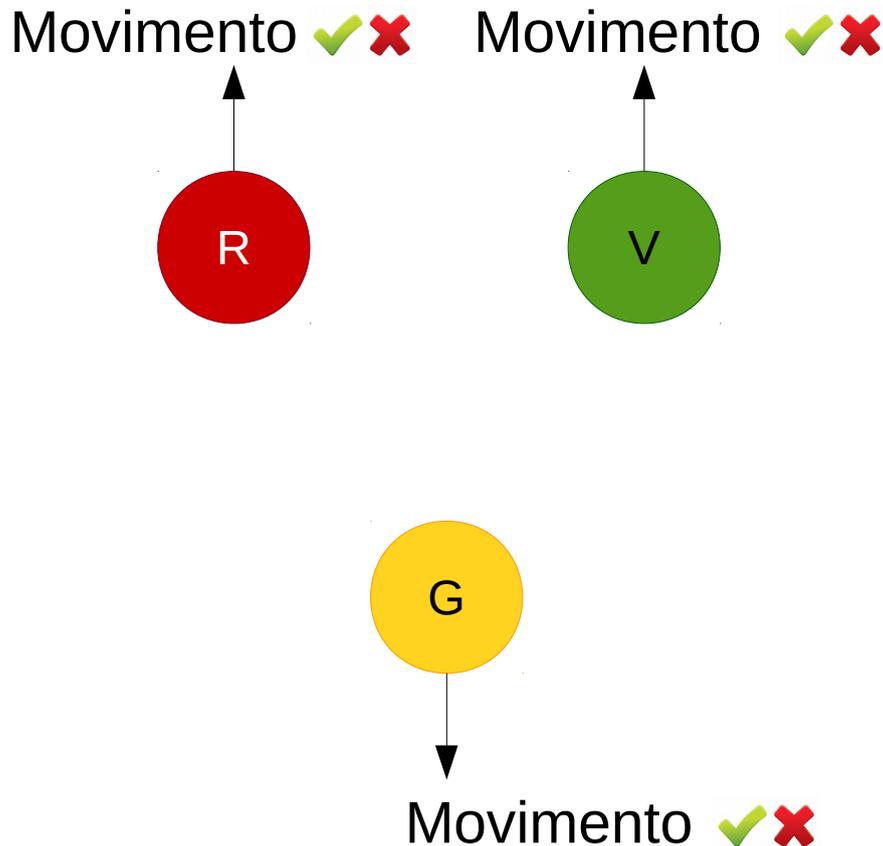


Matrice A

	R	G	V
R	0.8	0	0.2
G	0.4	0.5	0.1
V	0	0.3	0.7

Esercizio

Il seguente scenario potrebbe essere descritto dal seguente modello di Markov a stati nascosti:



Matrice B

	✗	✓
R	0.95	0.05
G	0.3	0.7
V	0.1	0.9

Esercizio

Probabilità di trovarsi all'istante iniziale nei diversi stati:

Vettore p

			
	0.5	0.1	0.4

$$\begin{bmatrix} 0.5 & 0.1 & 0.4 \end{bmatrix}$$

Esercizio

1. Costruire in Matlab le matrici A e B, e il vettore p, con i valori riportati nelle slide precedenti.
- Ricordarsi di includere nel path le funzioni contenute nello zip

```
>> pathtool
```

oppure

```
>> addpath(genpath(' ./HMM/ ' ))
```

Esercizio

A partire dalle matrici A e B, e il vettore p, generare una sequenza di 1000 stati e la corrispondente sequenza di osservazioni.

Idee:

- Estraggo il primo stato a partire dalle probabilità in p

```
>> stati(1) = extract_disc_state(p)
```

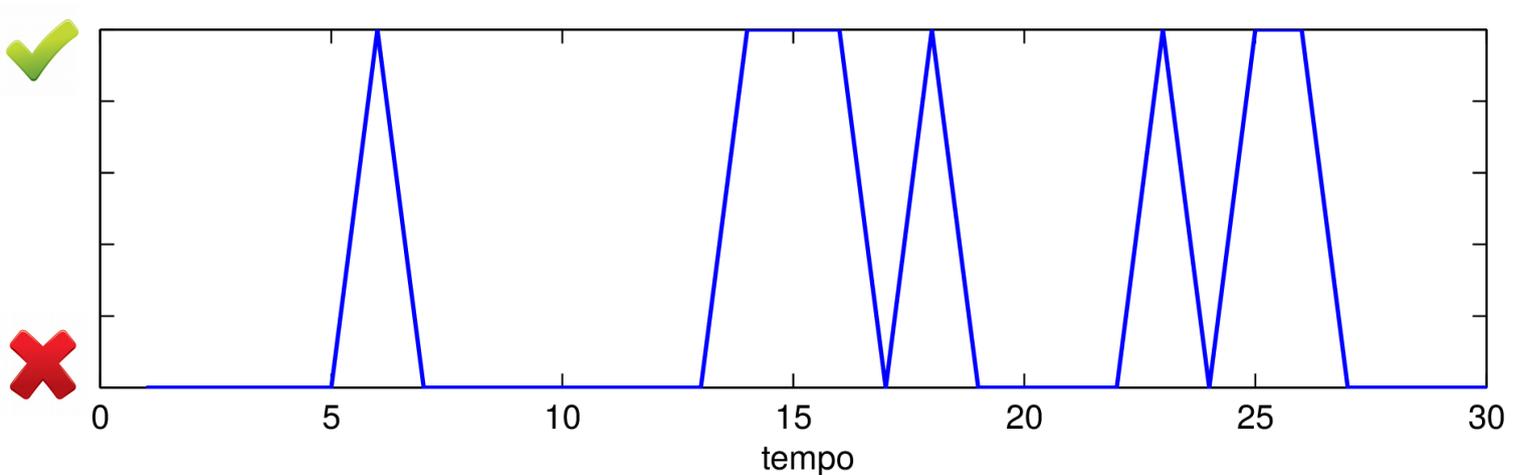
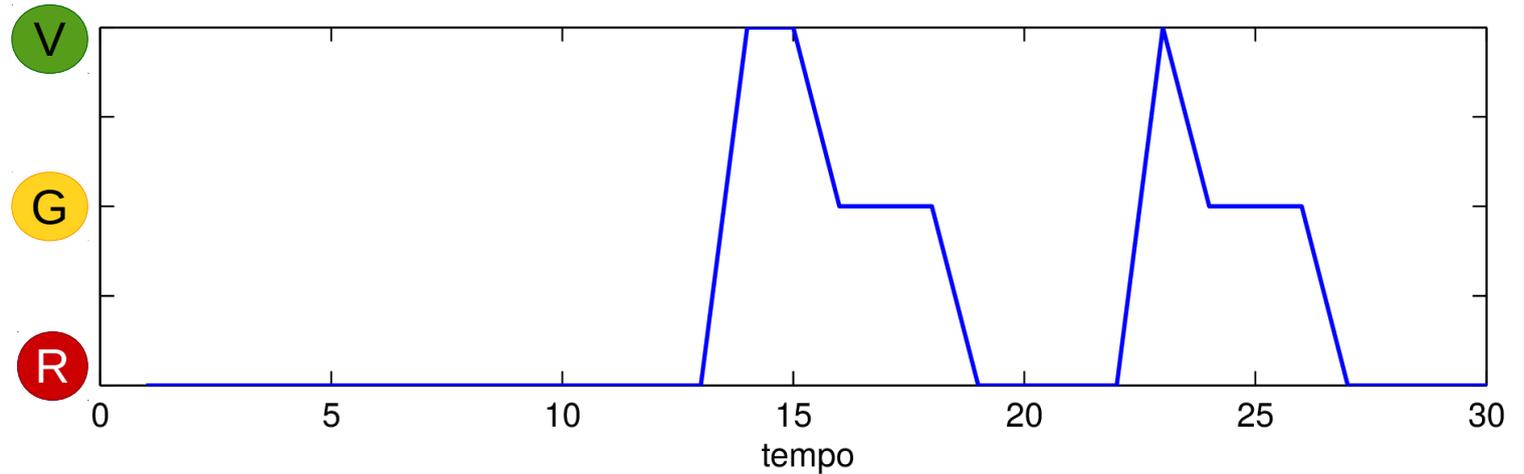
Esercizio

- Per generare lo stato al tempo $t+1$, posso usare sempre la funzione `extract_disc_state`:
 - Il vettore di input sarà la riga della matrice A corrispondente allo stato al tempo t
- Faccio un ragionamento simile per generare l'osservazione al tempo t , utilizzando però la matrice B

```
>> oss(1) = extract_disc_state(B(stati(1,:)))
```

Esercizio

- Plottare i primi 30 stati e le prime 30 osservazioni
- Intuitivamente, le osservazioni sono coerenti con gli stati?



Esercizio 2

- A questo punto considero semplicemente il vettore d'osservazioni `oss`
- Obiettivo: addestrare un HMM
- Primo: decido quanti stati dovranno esserci nell'HMM e quanti simboli ho a disposizione per l'osservazione

```
>> Q = 3; % numero di stati  
>> O = 2; % numero di simboli
```

Esercizio 2

- Poi: inizializzo a caso le matrici A, B, e p

```
>> % initial guess of parameters  
>> p1 = normalise(rand(Q,1));  
>> A1 = mk_stochastic(rand(Q,Q));  
>> B1 = mk_stochastic(rand(Q,0));
```

Esercizio 2

- Infine: addestro l'HMM

```
>> [LL, p1, A1, B1] = ...  
dhmm_em(oss, p1, A1, B1, 'max_iter', 10);
```

- Si riescono a ricostruire le matrici A, B, e p?
(attenzione all'identità degli stati)

Esercizio 3

- Considerare un nuovo HMM così specificato:

Matrice A

Matrice B

			
	0.6	0.1	0.3
	0.4	0.3	0.3
	0.2	0	0.8

		
	0.7	0.3
	0.2	0.8
	0	1

Vettore p

0.5	0.1	0.4
		

Esercizio 3

- Calcolare la likelihood delle osservazioni `oss` valutata rispetto:
 - 1. all'HMM addestrata nell'esercizio precedente
 - 2. alla seconda HMM appena definita

```
>> help dhmm_logprob
```

- Quale delle due likelihood è maggiore?

Esercizio 4

- Attraverso l'algoritmo di Viterbi, si può ricostruire la sequenza di stati del semaforo che ha portato alle osservazioni?

```
>> % In generale:  
>> obslik = mk_dhmm_obs_lik(oss, B);  
>> path = viterbi_path(p, A, obslik);
```

- A, B e p sono quelle stimate dopo l'addestramento (Es. 2)