



---

Corso di Laurea in Matematica, Università di Pisa.

# Alcune applicazioni di modelli di Markov nascosti.

Tesi di Laurea Triennale

*CANDIDATA:*  
Sara Ambrosi

*RELATORE:*  
Dott. Dario Trevisan

Anno Accademico 2019/2020



# Indice

<b>1</b>	<b>La teoria.</b>	<b>6</b>
1.1	Introduzione ai Modelli di Markov Nascosti. . . . .	6
1.2	Distribuzioni di filtraggio e smoothing. . . . .	12
1.3	Algoritmo di Baum-Welch. . . . .	19
1.4	Algoritmo di Viterbi. . . . .	23
1.5	Algoritmo EM per l'inferenza statistica. . . . .	25
1.6	Modelli non omogenei. . . . .	28
<b>2</b>	<b>Applicazione ai canali di trasmissione.</b>	<b>30</b>
2.1	Impiego dell'algoritmo di Viterbi. . . . .	32
2.2	Impiego dell'algoritmo EM. . . . .	37
<b>3</b>	<b>Un modello di tipo Galton-Watson.</b>	<b>42</b>
3.1	Modellizzazione. . . . .	42
3.2	Alcune simulazioni. . . . .	44
3.3	Algoritmo di Baum-Welch adattato al caso. . . . .	52
3.4	Un'ultima osservazione sull'inferenza. . . . .	57
<b>A</b>	<b>Teorema di Fubini-Tonelli per i nuclei di transizione.</b>	<b>59</b>
	<b>Bibliografia.</b>	

# Introduzione.

L'idea sviluppata in questa tesi è quella di studiare come possano essere applicati i modelli di Markov nascosti in alcuni contesti. Nello specifico sono stati applicati al caso dell'invio di un messaggio attraverso un canale di trasmissione affetto da rumore e alla diffusione di una malattia infettiva in una popolazione. La tesi è suddivisa in tre capitoli: nel primo è esposta la teoria necessaria a giustificare e sviluppare la parte sperimentale, negli altri due sono presentate le applicazioni descritte sopra.

In seguito si espone, capitolo per capitolo, quello che è stato fatto in breve:

## Capitolo 1

Nella prima sezione si danno le definizioni di modello di Markov omogeneo e modello di Markov nascosto e si mostra come si può ottenere la legge di quest'ultimo in termini dei cosiddetti nuclei di transizione e osservazione: un modello di Markov nascosto è un processo stocastico definito su uno spazio misurabile prodotto le cui componenti vengono denominate segnale e osservazione.

Nella seconda sezione si danno le definizioni di distribuzioni di filtraggio, smoothing e smoothing bivariato e si espongono alcune dimostrazioni finalizzate al loro calcolo.

Nella terza sezione si analizza il caso particolare in cui lo spazio degli stati del segnale sia finito e si arriva all'implementazione Matlab dell'algoritmo di Baum-Welch, che calcola le distribuzioni introdotte nella sezione precedente in questo caso particolare.

Nella quarta sezione, dove si mantiene l'ipotesi di spazio dei segnali finito, si arriva all'algoritmo di Viterbi il cui scopo è ricostruire la più probabile sequenza di segnali corrispondente ad una certa sequenza di osservazioni.

La quinta sezione del capitolo è dedicata all'inferenza statistica. Qui si espongono il metodo della massima verosimiglianza e l'algoritmo EM, che permettono di calcolare una stima dei parametri reali del modello sotto opportune ipotesi.

Nell'ultima sezione si danno le definizioni di processo di Markov e modello di Markov nascosto non omogenei e si fanno alcune considerazioni riguardando la teoria relativa a questi.

## Capitolo 2

Il capitolo inizia con la modellizzazione, tramite un modello di Markov nascosto, dell'invio di un messaggio (una stringa binaria) attraverso un canale disturbato.

Nella prima sezione è stata simulata la composizione del messaggio con una certa probabilità sul singolo bit di essere 0 e l'invio del messaggio con una certa probabilità di inviare un bit corretto. È stato poi affrontato il problema della ricostruzione del messaggio originario: ricostruire direttamente l'input a partire dall'output fa ottenere un messaggio originario errato a causa del fatto che il canale non è perfetto. È stato allora utilizzato l'algoritmo di Viterbi per ricostruire una stima dell'input originario.

Nella seconda sezione, supponendo di avere accesso al messaggio ricevuto e a dei possibili modelli di Markov nascosti che si adattano alla situazione e variano in base alla probabilità di inviare un bit corretto e alla probabilità con cui viene generato il bit 0, è stato affrontato il problema di ottenere una stima, tramite l'algoritmo EM, dei parametri veri.

## Capitolo 3

L'ultimo capitolo è suddiviso in quattro sezioni. Nella prima si dimostra di poter utilizzare un modello di Markov nascosto per modellizzare la diffusione di un virus tra la popolazione, tramite un processo di tipo Galton-Watson (nascita-morte a tempi discreti).

Nella seconda si fanno due simulazioni della diffusione del virus nel caso in cui ad un certo istante temporale varino i parametri del modello, ad esempio per l'introduzione di misure di contenimento del contagio, e se ne studia il comportamento.

Nella terza sezione viene implementato l'algoritmo di Baum-Welch per calcolare le distribuzioni di filtraggio, smoothing e smoothing bivariato e nell'ultima sezione si fanno alcune osservazioni su quest'ultimo e sull'inferenza.

# Capitolo 1

## La teoria.

### 1.1 Introduzione ai Modelli di Markov Nascosti.

Introduciamo alcune definizioni e proprietà fondamentali per comprendere il lavoro che andremo a fare nelle prossime sezioni.

Lavoriamo su uno spazio di probabilità  $(\Omega, \mathcal{G}, \mathbf{P})$ .

**Definizione 1.1** (Nucleo di transizione). Un nucleo di transizione tra due spazi misurabili  $(E, \mathcal{E})$  e  $(F, \mathcal{F})$  è una funzione  $P : E \times \mathcal{F} \rightarrow \mathbb{R}_+$  t.c.

1.  $A \mapsto P(x, A)$  è una misura e  $P(x, F) = 1$  per ogni  $x \in E$ ;
2.  $x \mapsto P(x, A)$  è misurabile per ogni  $A \in \mathcal{F}$ .

**Definizione 1.2** (Processo di Markov omogeneo). Un processo di Markov omogeneo è un processo stocastico  $(X_k)_{k \geq 0}$  a valori in  $(E, \mathcal{E})$  che rispetta la seguente proprietà:

Esiste un nucleo di transizione  $P : E \times \mathcal{E} \rightarrow [0, 1]$  t.c. per ogni  $k \geq 0$  e per ogni  $A \in \mathcal{E}$  si ha

$$\mathbf{P}(X_{k+1} \in A | X_0, \dots, X_k) = P(X_k, A)$$

**Definizione 1.3** (Misura iniziale di un processo di Markov omogeneo). Dato un processo di Markov omogeneo come sopra, definiamo misura iniziale del processo la funzione  $\mu : \mathcal{E} \rightarrow [0, 1]$  definita come  $\mu(A) = \mathbf{P}(X_0 \in A)$  con  $A \in \mathcal{E}$ .

Osserviamo ora che conoscendo la misura iniziale  $\mu$  e il nucleo di transizione  $P$  di un processo di Markov omogeneo possiamo riacavarne la legge. Vale infatti il seguente lemma:

**Lemma 1.4** (Legge di un processo di Markov omogeneo). Sia  $(X_k)_{k \geq 0}$  un processo di Markov omogeneo su  $(E, \mathcal{E})$  con nucleo di transizione  $P$  e misura iniziale  $\mu$ . Allora per ogni funzione misurabile e limitata  $f : E^{k+1} \rightarrow \mathbb{R}$  si ha:

$$E[f(X_0, \dots, X_k)] = \int f(x_0, \dots, x_k) P(x_{k-1}, dx_k) \dots P(x_0, dx_1) \mu(dx_0)^1$$

<sup>1</sup>Il teorema di Fubini-Tonelli per i nuclei (in appendice) iterato  $k$  volte dà senso all'integrale.

*Dimostrazione.* Ci limitiamo a dimostrare la tesi per funzioni della forma

$$f(x_0, \dots, x_k) = f_0(x_0) \cdots f_k(x_k)$$

Usando il teorema delle classi monotone si ottiene poi il caso generale. Sviluppriamo allora il membro di sinistra in questo caso particolare.

$$\begin{aligned} E[f_0(X_0) \cdots f_k(X_k)] &= E[f_0(X_0) \cdots f_{k-1}(X_{k-1}) E[f_k(X_k) | X_0, \dots, X_{k-1}]] \\ &= E[f_0(X_0) \cdots f_{k-1}(X_{k-1}) \int f_k(x_k) P(X_{k-1}, dx_k)] \\ &= E[f_0(X_0) \cdots f_{k-2}(X_{k-2}) \cdot \\ &\quad \cdot E[f_{k-1}(X_{k-1}) \int f_k(x_k) P(X_{k-1}, dx_k) | X_0, \dots, X_{k-2}]] \\ &= E[f_0(X_0) \cdots f_{k-2}(X_{k-2}) \cdot \\ &\quad \cdot E[\int f_{k-1}(x_{k-1}) f_k(x_k) P(x_{k-1}, dx_k) P(X_{k-2}, dx_{k-1})]] \end{aligned}$$

Procedendo in questo modo si ottiene:

$$\begin{aligned} &E[f_0(X_0) \cdots f_{k-2}(X_{k-2}) \cdot \\ &\quad \cdot E[\int f_{k-1}(x_{k-1}) f_k(x_k) P(x_{k-1}, dx_k) P(X_{k-2}, dx_{k-1})]] = \\ &= E[f_0(X_0) \int f_1(x_1) \cdots f_k(x_k) P(x_{k-1}, dx_k) \cdots P(X_0, dx_1)] \\ &= \int f_0(x_0) \cdots f_k(x_k) P(x_{k-1}, dx_k) \cdots P(x_0, dx_1) \mu(dx_0) \end{aligned}$$

□

Veniamo ora ai Modelli di Markov Nascosti, diamone la definizione:

**Definizione 1.5** (Modello di Markov Nascosto). Un processo stocastico  $(X_k, Y_k)_{k \geq 0}$  a valori in  $(E \times F, \mathcal{E} \otimes \mathcal{F})$  è detto modello di Markov nascosto se:

1. esistono due nuclei di transizione  $P : E \times \mathcal{E} \rightarrow [0, 1]$  e  $\Phi : E \times \mathcal{F} \rightarrow [0, 1]$  tali che per ogni  $g : E \times F \rightarrow \mathbb{R}$  misurabile e limitata si abbia:

$$E[g(X_{k+1}, Y_{k+1}) | X_0, Y_0, \dots, X_k, Y_k] = \int_{E \times F} g(x, y) \Phi(x, dy) P(X_k, dx)$$

2. esiste una misura di probabilità  $\mu$  su  $E$  t.c.

$$E[g(X_0, Y_0)] = \int_{E \times F} g(x, y) \Phi(x, dy) \mu(dx)$$

per ogni  $g : E \times F \rightarrow \mathbb{R}$  misurabile e limitata.

Chiamiamo  $\mu$  misura iniziale,  $P$  nucleo di transizione e  $\Phi$  nucleo di osservazione del modello.

Dato un modello di Markov nascosto come sopra, ci riferiremo a  $(X_k)_{k \geq 0}$  parlando di segnali del modello e ad  $(Y_k)_{k \geq 0}$  parlando di osservazioni.

Vediamo ora il seguente lemma che ci permette di capire qualcosa di più sulla natura di un modello di Markov nascosto.

**Lemma 1.6** (Proprietà di un modello di Markov nascosto). *Sia  $(X_k, Y_k)_{k \geq 0}$  un modello di Markov nascosto su  $(E \times F, \mathcal{E} \otimes \mathcal{F})$  con nucleo di transizione  $P$ , nucleo di osservazione  $\Phi$  e misura iniziale  $\mu$ . Valgono i seguenti fatti:*

1.  $(X_k, Y_k)_{k \geq 0}$  è un processo di Markov omogeneo;
2.  $(X_k)_{k \geq 0}$  è un processo di Markov omogeneo con nucleo  $P$ ;
3. le variabili  $Y_0, \dots, Y_k$  sono condizionalmente indipendenti dalla  $\sigma$ -algebra generata da  $X_0, \dots, X_k$ :

$$P(Y_0 \in A_0, \dots, Y_k \in A_k | X_0, \dots, X_k) = \prod_{i=0}^k \Phi(X_i, A_i).$$

*Dimostrazione.* 1. Per mostrare che il modello di Markov nascosto dato è un processo di Markov omogeneo si deve verificare l'esistenza di un nucleo di transizione  $N : (E \times F) \times (\mathcal{E} \otimes \mathcal{F}) \rightarrow [0, 1]$  che rispetta la seguente proprietà:

$$\forall C \in (\mathcal{E} \otimes \mathcal{F}), \forall k \geq 0,$$

$$P((X_{k+1}, Y_{k+1}) \in C | X_0, Y_0, \dots, X_k, Y_k) = N((X_k, Y_k), C)$$

Osservazione. Basta mostrarlo sui rettangoli misurabili (quindi  $C = A \times B$ ,  $A \in \mathcal{E}$  e  $B \in \mathcal{F}$ ) perché questa famiglia è un'algebra e genera la  $\sigma$ -algebra  $\mathcal{E} \otimes \mathcal{F}$  e dunque una volta mostrata l'esistenza della probabilità di destra sui rettangoli misurabili si può usare il teorema di Carathéodory per dire che questa si estende su tutta la  $\sigma$ -algebra e necessariamente coincide con la probabilità condizionata.<sup>2</sup>

---

<sup>2</sup>In realtà si dovrebbe far vedere che, per ogni  $C \in \mathcal{E} \otimes \mathcal{F}$ ,  $N$  non dipende dalle componenti del processo per  $i \leq k$ : per far vedere questo si può scrivere  $C$  come unione crescente di insiemi dati dall'unione di rettangoli misurabili e usare su  $N$  la proprietà di passaggio al limite.



Procediamo con la verifica sviluppando il membro di sinistra:

$$\begin{aligned}
\mathbf{P}((X_{k+1}, Y_{k+1}) \in A \times B | X_0, Y_0, \dots, X_k, Y_k) &= \\
&= \mathbf{E}[I_{(A \times B)}(X_{k+1}, Y_{k+1}) | X_0, Y_0, \dots, X_k, Y_k] \\
&= \int_{E \times F} I_{(A \times B)}(x, y) \Phi(x, dy) P(X_k, dx) \\
&= \int_{A \times B} \Phi(x, dy) P(X_k, dx) \\
&= \int_A \left[ \int_B \Phi(x, dy) \right] P(X_k, dx) \\
&= \int_A \Phi(x, B) P(X_k, dx).
\end{aligned}$$

Questa è una funzione soltanto di  $A \times B$  e  $X_k$  a valori in  $[0,1]$ .  
Inoltre:

$$\begin{aligned}
N((X_k, Y_k), E \times F) &= \int_E \Phi(x, F) P(X_k, dx) \\
&= \int_E P(X_k, dx) \\
&= P(X_k, E) \\
&= 1.
\end{aligned}$$

La verifica delle proprietà di nucleo segue da quanto fatto in appendice.

2. Per mostrare che  $(X_k)_{k \geq 0}$  è un processo di Markov omogeneo di nucleo  $P$  si deve far vedere che vale:

$$\forall A \in \mathcal{E}, \forall k \geq 0,$$

$$\mathbf{P}(X_{k+1} \in A | X_0, \dots, X_k) = P(X_k, A).$$

Procediamo sviluppando il membro di sinistra:

$$\begin{aligned}
\mathbf{P}(X_{k+1} \in A | X_0, \dots, X_k) &= \mathbf{E}[I_A(X_{k+1}) | X_0, \dots, X_k] \\
&= \mathbf{E}[I_{A \times E}(X_{k+1}, Y_{k+1}) | X_0, \dots, X_k]
\end{aligned}$$

Osservazione.

$$\sigma(X_0, \dots, X_k) \subseteq \sigma(X_0, Y_0, \dots, X_k, Y_k)$$

Per le proprietà della speranza condizionale:

$$\begin{aligned}
\mathbf{E}[I_{A \times E}(X_{k+1}, Y_{k+1}) | X_0, \dots, X_k] &= \\
&= \mathbf{E}[\mathbf{E}[I_{A \times E}(X_{k+1}, Y_{k+1}) | X_0, Y_0, \dots, X_k, Y_k] | X_0, \dots, X_k] \\
&= \mathbf{E}\left[\int_A \Phi(x, E) P(X_k, dx) \mid X_0, \dots, X_k\right] \\
&= \mathbf{E}\left[\int_A P(X_k, dx) \mid X_0, \dots, X_k\right] \\
&= \mathbf{E}[P(X_k, A) | X_0, \dots, X_k] = P(X_k, A)
\end{aligned}$$

(L'ultimo passaggio è giustificato dalla  $\sigma(X_0, \dots, X_k)$ -misurabilità di  $X_k$ ).

3. Riscriviamo l'uguaglianza del punto 3 come:

$$\mathbf{E}\left[\prod_{i=0}^k I_{A_i}(Y_i) \mid X_0, \dots, X_k\right] = \prod_{i=0}^k \Phi(X_i, A_i)$$

Questo equivale a dire che per ogni  $B_0, \dots, B_k$  insiemi in  $\varepsilon$  si ha:

$$\mathbf{E}\left[\prod_{j=0}^k I_{B_j}(X_j) \prod_{i=0}^k I_{A_i}(Y_i)\right] = \mathbf{E}\left[\prod_{j=0}^k I_{B_j}(X_j) \prod_{i=0}^k \Phi(X_i, A_i)\right]$$

Procediamo per induzione.

i) Passo base:  $k=0$ .

$$\mathbf{E}[I_{B_0}(X_0)I_{A_0}(Y_0)] = \int_{E \times F} I_{B_0}(x)I_{A_0}(y)\Phi(x, dy)\mu^*(dx)$$

Osserviamo che  $\mu^*$  misura iniziale di  $(X_k, Y_k)_{k \geq 0}$  e  $\mu$  misura iniziale di  $(X_k)_{k \geq 0}$  sono uguali.

$$\begin{aligned} \mathbf{E}[I_{B_0}(X_0)I_{A_0}(Y_0)] &= \int_E I_{B_0}(x)\Phi(x, A_0)\mu(dx) \\ &= \mathbf{E}[I_{B_0}(X_0)\Phi(X_0, A_0)]. \end{aligned}$$

ii) Passo induttivo: supponiamo l'uguaglianza sia vera per  $k$  e mostriamolo per  $k+1$ .

$$\begin{aligned} \mathbf{E}\left[\prod_{j=0}^{k+1} I_{B_j}(X_j) \prod_{i=0}^{k+1} I_{A_i}(Y_i)\right] &= \\ &= \int_{(E \times F)^k} \left(\prod_{j=0}^k I_{B_j}(x_j)I_{A_j}(y_j)\Phi(x_j, dy_j)\right) \left[\int_{B_{k+1} \times A_{k+1}} \Phi(x_{k+1}, dy_{k+1})P(x_k, dx_{k+1})\right] \cdot \\ &\cdot P(x_{k-1}, dx_k) \cdots P(x_0, dx_1)\mu(dx_0) = \end{aligned}$$

Il membro tra parentesi quadre sarà una funzione  $g(x_k)$

$$\begin{aligned} &= \int_{(E \times F)^k} \left(\prod_{j=0}^k I_{B_j}(x_j)I_{A_j}(y_j)\Phi(x_j, dy_j)\right) g(x_k)P(x_{k-1}, dx_k) \cdots P(x_0, dx_1)\mu(dx_0) \\ &= \mathbf{E}\left[g(X_k) \prod_{j=0}^k I_{B_j}(X_j) \prod_{i=0}^k I_{A_i}(Y_i)\right] = \end{aligned}$$

Sfrutto l'ipotesi induttiva.

$$\begin{aligned}
&= \mathbf{E}[g(X_k) \prod_{j=0}^k I_{B_j}(X_j) \prod_{i=0}^k \Phi(X_i, A_i)] \\
&= \int_{(E \times F)^k} g(x_k) \prod_{j=0}^k I_{B_j}(x_j) \Phi(x_j, A_j) P(x_{k-1}, dx_k) \cdots P(x_0, dx_1) \mu(dx_0).
\end{aligned}$$

□

In particolare questo lemma e il lemma 1.4 ci permettono di ottenere le distribuzioni finito dimensionali di un modello di Markov nascosto  $(X_k, Y_k)_{k \geq 0}$ . Vale il seguente:

**Corollario 1.7** (Legge di un modello di Markov nascosto). *Dato un modello di Markov nascosto  $(X_k, Y_k)_{k \geq 0}$  su  $(E \times F, \mathcal{E} \otimes \mathcal{F})$ , per ogni funzione misurabile e limitata  $f : (E \times F)^{k+1} \rightarrow \mathbb{R}$  si ha:*

$$\begin{aligned}
E[f(X_0, Y_0, \dots, X_k, Y_k)] &= \int f(x_0, y_0, \dots, x_k, y_k) \Phi(x_k, dy_k) P(x_{k-1}, dx_k) \cdots \\
&\quad \cdots \cdot \Phi(x_1, dy_1) P(x_0, dx_1) \Phi(x_0, dy_0) \mu(dx_0)
\end{aligned}$$

Diamo un'ultima definizione riguardante le osservazioni  $(Y_k)_{k \geq 0}$  di un modello di Markov nascosto.

**Definizione 1.8** (Osservazioni non degeneri). Sia  $(X_k, Y_k)_{k \geq 0}$  un modello di Markov nascosto su  $(E \times F, \mathcal{E} \otimes \mathcal{F})$  con nucleo di osservazione  $\Phi$ . Il modello è detto avere osservazioni non degeneri se il nucleo di osservazione è della forma:

$$\Phi(x, B) = \int I_B(z) \gamma(x, z) \varphi(dz), \quad x \in E, B \in \mathcal{F}$$

dove  $\gamma : E \times F \rightarrow ]0, \infty[$  è una funzione misurabile, che chiameremo densità di osservazione, e  $\varphi$  è una misura di probabilità su  $F$ .

*Osservazione 1.9.* Questa definizione è particolarmente importante perché ci permette di applicare il Teorema di Bayes, enunciato nella prossima sezione. Inoltre, il fatto che  $\gamma$  sia strettamente positiva ci permette di implementare gli algoritmi che vedremo nelle sezioni successive senza opportuni aggiustamenti: nell'algoritmo di Baum-Welch, ad esempio, capita di dividere per numeri che vengono moltiplicati per  $\gamma$  e senza l'ipotesi di positività potrebbe capitare di dividere per zero. Nell'ultimo capitolo è presente un caso nel quale si deve aggiungere del rumore per ovviare a questo problema.

## 1.2 Distribuzioni di filtraggio e smoothing.

Iniziamo la sezione con una definizione.

**Definizione 1.10** (Distribuzione condizionata). Siano  $X$  una variabile aleatoria a valori in  $(E, \mathcal{E})$  e  $Y$  una variabile aleatoria a valori in  $(B, \mathcal{B})$  definite sullo spazio di probabilità  $(\Omega, \mathcal{G}, \mathbf{P})$ . Chiamo distribuzione condizionata di  $X$  data  $Y$  un nucleo di transizione  $P_{X|Y} : B \times \mathcal{E} \rightarrow [0, 1]$  che soddisfa:

$$E[f(X)|Y] = \int f(x)P_{X|Y}(Y, dx)$$

per ogni  $f : E \rightarrow \mathbb{R}$  limitata e misurabile.

Date due variabili aleatorie  $X$  e  $Y$  come sopra, vediamo un teorema che ci permette di ottenere la distribuzione condizionata di  $X$  rispetto ad  $Y$  sotto un'ipotesi aggiuntiva.

**Teorema 1.11** (Teorema di Bayes). *Siano  $X$  una v.a. a valori in  $(E, \mathcal{E})$  e  $Y$  una v.a. a valori in  $(B, \mathcal{B})$  definite su uno spazio di probabilità  $(\Omega, \mathcal{G}, \mathbf{P})$ . Supponiamo esistano una funzione misurabile  $\gamma : E \times B \rightarrow ]0, \infty[$ , una misura di probabilità  $\mu_X$  su  $E$  e una misura di probabilità  $\mu_Y$  su  $B$  tali che per ogni funzione misurabile e limitata  $f$  si abbia:*

$$E[f(X, Y)] = \int f(x, y)\gamma(x, y)\mu_X(dx)\mu_Y(dy)$$

Allora  $\forall A \in \mathcal{E}$  e  $\forall y \in B$  si ha che:

$$P_{X|Y}(y, A) = \frac{\int I_A(x)\gamma(x, y)\mu_X(dx)}{\int \gamma(x, y)\mu_X(dx)}$$

è la distribuzione condizionata di  $X$  rispetto ad  $Y$ .

*Dimostrazione.* Dobbiamo far vedere che per ogni  $A \in \mathcal{E}$  abbiamo  $P_{X|Y}(Y, A) = P(X \in A|Y)$ . Questo è equivalente a dire che per ogni  $A \in \mathcal{E}$  e per ogni  $C \in \mathcal{B}$  si ha  $E[P_{X|Y}(Y, A)I_C(Y)] = E[I_A(X)I_C(Y)]$ .

Sviluppiamo il membro di sinistra:

$$\begin{aligned} E[P_{X|Y}(Y, A)I_C(Y)] &= E\left[\frac{\int I_A(x')I_C(Y)\gamma(x', Y)\mu_X(dx')}{\int \gamma(x', Y)\mu_X(dx')}\right] \\ &= \int \frac{\int I_A(x')I_C(y)\gamma(x', y)\mu_X(dx')}{\int \gamma(x', y)\mu_X(dx')} \gamma(x, y)\mu_X(dx)\mu_Y(dy) \\ &= \int I_A(x')I_C(y)\gamma(x', y)\mu_X(dx')\mu_Y(dy) \\ &= E[I_A(X)I_C(Y)]. \end{aligned}$$

□

Supponiamo ora di avere un modello di Markov nascosto  $(X_k, Y_k)_{k \geq 0}$  con spazio dei segnali  $(E, \mathcal{E})$  e spazio delle osservazioni  $(F, \mathcal{F})$ , nucleo di transizione  $P$ , nucleo di osservazione  $\Phi$  e misura iniziale  $\mu$ . Supponiamo anche che le osservazioni siano non degeneri e indichiamo con  $\gamma$  la densità delle osservazioni e con  $\varphi$  la misura di probabilità come in definizione 1.8.

Concentriamoci sul problema di calcolare le distribuzioni condizionali del tipo  $P_{X_k|Y_0, \dots, Y_n}$  con  $k, n \geq 0$ . Indicheremo per brevità  $P_{X_k|Y_0, \dots, Y_n}$  con  $\pi_{k|n}$ . Distinguiamo tre casi :

1.  $k=n$  problema del filtraggio;
2.  $k < n$  problema dello smoothing;
3.  $k > n$  problema della previsione.

Ai fini della tesi a noi interessano i primi due. Il terzo è però importante in molti problemi e dunque vedremo anche quello.

Iniziamo con lo scrivere la legge finito dimensionale del modello, che possiamo ottenere dal corollario 1.7. Data  $f : E^{k+1} \times F^{k+1} \rightarrow \mathbb{R}$  misurabile e limitata si ha:

$$\begin{aligned} E[f(X_0, Y_0, \dots, X_n, Y_n)] &= \\ &= \int f(x_0, y_0, \dots, x_n, y_n) \gamma(x_0, y_0) \cdots \gamma(x_n, y_n) \varphi(dy_0) \cdots \varphi(dy_n) \times \\ &\times P(x_{n-1}, dx_n) \cdots P(x_0, dx_1) \mu(dx_0) \end{aligned}$$

L'idea è quella di combinare quest'espressione con il teorema di Bayes per ottenere le distribuzioni che cerchiamo.

Iniziamo con una definizione.

**Definizione 1.12** (Distribuzione di filtraggio non normalizzata). Fissato  $k \in \mathbb{N}$  chiamiamo distribuzione di filtraggio non normalizzata la funzione  $\sigma_k : F^{k+1} \times \mathcal{E} \rightarrow \mathbb{R}_+$  definita come:

$$\begin{aligned} \sigma_k(y_0, \dots, y_k, A) &= \\ &= \int I_A(x_k) \gamma(x_0, y_0) \cdots \gamma(x_k, y_k) P(x_{k-1}, dx_k) \cdots P(x_0, dx_1) \mu(dx_0) \end{aligned}$$

per ogni  $y_0, \dots, y_k \in F$  e  $A \in \mathcal{E}$ .

Mostriamo ora che la normalizzazione di questa funzione coincide con la distribuzione di filtraggio  $\pi_k := \pi_{k|k}$ .

**Teorema 1.13** (Distribuzione di filtraggio). Per ogni  $y_0, \dots, y_k \in F$  e ogni  $A \in \mathcal{E}$  si ha:

$$\pi_k(y_0, \dots, y_k, A) = \frac{\sigma_k(y_0, \dots, y_k, A)}{\sigma_k(y_0, \dots, y_k, E)}$$

*Dimostrazione.* Vogliamo applicare il teorema di Bayes. Mantenendo la notazione usata nel teorema definiamo la misura di probabilità  $\mu_Y$  su  $F^{k+1}$  come la misura prodotto

$$\mu_Y(dy_0, \dots, dy_k) = \varphi(dy_0) \cdots \varphi(dy_k)$$

E la misura di probabilità  $\mu_X$  su  $E^{k+1}$  come

$$\mu_X(dx_0, \dots, dx_k) = P(x_{k-1}, dx_k) \cdots P(x_0, dx_1)\mu(dx_0)$$

Abusando di notazione definiamo la funzione  $\gamma$  come

$$\gamma(x_0, \dots, x_k, y_0, \dots, y_k) = \gamma(x_0, y_0) \cdots \gamma(x_k, y_k)$$

Utilizzando la formula di Bayes otteniamo:

$$\begin{aligned} \int f(x_0, \dots, x_k) P_{X_0, \dots, X_k | Y_0, \dots, Y_k}(y_0, \dots, y_k, dx_0, \dots, dx_k) &= \\ &= \frac{\int f(x_0, \dots, x_k) \gamma(x_0, \dots, x_k, y_0, \dots, y_k) \mu_X(dx_0, \dots, dx_k)}{\int \gamma(x_0, \dots, x_k, y_0, \dots, y_k) \mu_X(dx_0, \dots, dx_k)} \end{aligned}$$

La tesi segue da:

$$\begin{aligned} \int f(x) \tau_k(y_0, \dots, y_k, dx) &= \int f(x_k) P_{X_k | Y_0, \dots, Y_k}(y_0, \dots, y_k, dx_k) \\ &= \int f(x_k) P_{X_0, \dots, X_k | Y_0, \dots, Y_k}(y_0, \dots, y_k, dx_0, \dots, dx_k) \\ &= \frac{\int f(x_k) \gamma(x_0, \dots, x_k, y_0, \dots, y_k) \mu_X(dx_0, \dots, dx_k)}{\int \gamma(x_0, \dots, x_k, y_0, \dots, y_k) \mu_X(dx_0, \dots, dx_k)} \\ &= \frac{\int f(x_k) \sigma_k(y_0, \dots, y_k, dx_k)}{\int \sigma_k(y_0, \dots, y_k, dx_k)} \end{aligned}$$

□

**Teorema 1.14** (Ricorsione per la distribuzione di filtraggio non normalizzata.).  
La distribuzione di filtraggio non normalizzata può essere calcolata ricorsivamente come:

$$\begin{cases} \sigma_k(y_0, \dots, y_k, A) = \int I_A(x) \gamma(x, y_k) P(x', dx) \sigma_{k-1}(y_0, \dots, y_{k-1}, dx') \\ \sigma_0(y_0, A) = \int I_A(x) \gamma(x, y_0) \mu(dx) \end{cases}$$

*Dimostrazione.* È una conseguenza diretta delle definizioni. □

Possiamo quindi affermare che:

**Corollario 1.15** (Ricorsione del filtraggio.). *La distribuzione di filtraggio normalizzata può essere calcolata ricorsivamente come:*

$$\begin{cases} \pi_k(y_0, \dots, y_k, A) = \frac{\int I_A(x) \gamma(x, y_k) P(x', dx) \pi_{k-1}(y_0, \dots, y_{k-1}, dx')}{\int \gamma(x, y_k) P(x', dx) \pi_{k-1}(y_0, \dots, y_{k-1}, dx')} \\ \pi_0(y_0, A) = \frac{\int I_A(x) \gamma(x, y_0) \mu(dx)}{\int \gamma(x, y_0) \mu(dx)} \end{cases}$$

Passiamo ora al problema dello smoothing.

Iniziamo con una definizione.

**Definizione 1.16** (Densità di smoothing non normalizzata). Per ogni  $0 \leq k < n$  definiamo densità di smoothing non normalizzata la funzione  $\beta_{k|n} : E \times F^{n-k} \rightarrow ]0, \infty[$  definita come:

$$\begin{aligned} \beta_{k|n}(x_k, y_{k+1}, \dots, y_n) &= \\ &= \int \gamma(x_{k+1}, y_{k+1}) \cdots \gamma(x_n, y_n) P(x_{n-1}, dx_n) \cdots P(x_k, dx_{k+1}) \end{aligned}$$

Per ogni  $y_{k+1}, \dots, y_n \in F$  e  $x_k \in E$ .

Sempre tramite la formula di Bayes dimostriamo il seguente teorema.

**Teorema 1.17** (Distribuzione di smoothing.). *La distribuzione di smoothing  $\pi_{k|n}$  vale:*

$$\pi_{k|n}(y_0, \dots, y_n, A) = \frac{\int I_A(x) \beta_{k|n}(x, y_0, \dots, y_n) \sigma_k(y_0, \dots, y_k, dx)}{\int \beta_{k|n}(x, y_{k+1}, \dots, y_n) \sigma_k(y_0, \dots, y_k, dx)}$$

Per ogni  $A \in \mathcal{E}$  e  $y_0, \dots, y_n \in F$ .

*Dimostrazione.* Usando la stessa notazione del teorema 1.13. abbiamo:

$$\begin{aligned} \int f(x) \pi_{k|n}(y_0, \dots, y_n, dx) &= \int f(x_k) P_{X_k|Y_0, \dots, Y_n}(y_0, \dots, y_n, dx_k) \\ &= \int f(x_k) P_{X_0, \dots, X_n|Y_0, \dots, Y_n}(y_0, \dots, y_n, dx_0, \dots, dx_n) \\ &= \frac{\int f(x_k) \gamma(x_0, \dots, x_k, y_0, \dots, y_k) \mu_X(dx_0, \dots, dx_k)}{\int \gamma(x_0, \dots, x_k, y_0, \dots, y_k) \mu_X(dx_0, \dots, dx_k)} \\ &= \frac{\int f(x_k) \beta_{k|n}(x_k, y_{k+1}, \dots, y_n) \sigma_k(y_0, \dots, y_k, dx_k)}{\int \beta_{k|n}(x_k, y_{k+1}, \dots, y_n) \sigma_k(y_0, \dots, y_k, dx_k)} \end{aligned}$$

□

**Teorema 1.18** (Ricorsione per la densità di smoothing non normalizzata.). *La densità di smoothing non normalizzata può essere calcolata ricorsivamente come:*

$$\begin{cases} \beta_{k|n}(x, y_{k+1}, \dots, y_n) = \int \beta_{k+1|n}(x', y_{k+2}, \dots, y_n) \gamma(x', y_{k+1}) P(x, dx') \\ \beta_{n|n} = 1 \end{cases}$$

*Dimostrazione.* Si verifica facilmente dalle definizioni.  $\square$

Vediamo ora un'altra distribuzione che chiameremo distribuzione di smoothing bivariato. Per  $l < n$  chiamiamo  $\pi_{l,l+1|n} = P_{X_l, X_{l+1} | Y_0, \dots, Y_n}$ . Omettiamo per scorrevolezza le dipendenze da  $y_0, \dots, y_n$  dove sono note. Vale il seguente:

**Teorema 1.19** (Distribuzione di smoothing bivariato.).

$$\begin{aligned} \pi_{l,l+1|n}(A \times B) &= \\ &= \frac{\int I_A(x_l) I_B(x_{l+1}) \beta_{l+1|n}(x_{l+1}) \gamma(x_{l+1}, y_{l+1}) P(x_l, dx_{l+1}) \sigma_l(dx_l)}{\int \beta_{l+1|n}(x_{l+1}) \sigma_{l+1}(dx_{l+1})} \end{aligned}$$

*Dimostrazione.* Chiamiamo  $X = (X_0, \dots, X_n)$  e  $Y = (Y_0, \dots, Y_n)$  e facciamo lo stesso con  $x$  e  $y$ . Osserviamo che data una funzione  $f(X, Y)$  si ha  $\mathbf{E}[f(X, Y)] = \int f(x, y) \gamma(x, y) \mu_X(dx) \mu_Y(dy)$  con la notazione del teorema 1.11. e quindi per la formula di Bayes conosciamo il valore di  $P_{X|Y}$ .

Si ha quindi:

$$\begin{aligned} \int f(x_{l+1}, x_l) \pi_{l,l+1|n}(dx_{l+1}, dx_l) &= \\ &= \int f(x_{l+1}, x_l) P_{X_{l+1}, X_l | Y}(dx_{l+1}, dx_l) \\ &= \int f(x_{l+1}, x_l) P_{X_{l+1}, \dots, X_0 | Y}(dx_{l+1}, dx_l, \dots, dx_0) \\ &= \int f(x_{l+1}, x_l) \frac{\gamma(x, y) \mu_X(dx)}{\int \gamma(x, y) \mu_X(dx)} \\ &= \frac{\int f(x_{l+1}, x_l) \gamma(x_{l+1}, y_{l+1}) P(x_l, dx_{l+1}) \beta_{l+1|n}(x_{l+1}) \sigma_l(dx_l)}{\int \beta_{l+1|n}(x_{l+1}) \sigma_{l+1}(dx_{l+1})} \end{aligned}$$

$\square$

**Teorema 1.20** (Ricorsione per lo smoothing bivariato.). *Diamo ora la seguente definizione ricorsiva:*

$$\begin{cases} \tilde{\beta}_{k|n}(x, y_0, \dots, y_n) = \frac{\int \tilde{\beta}_{k+1|n}(x', y_0, \dots, y_n) \gamma(x', y_{k+1}) P(x, dx')}{\int \gamma(x', y_k) P(x, dx') \pi_{k-1}(y_0, \dots, y_{k-1}, dx)} \\ \tilde{\beta}_{n|n}(x, y_0, \dots, y_n) = \frac{1}{\int \gamma(x', y_n) P(x, dx') \pi_{n-1}(y_0, \dots, y_{n-1}, dx)} \end{cases}$$

*Dimostriamo che si può scrivere la distribuzione di smoothing bivariata in forma normalizzata come:*

$$\begin{aligned} \pi_{l,l+1|n}(A \times B) &= \\ &= \int I_A(x_l) I_B(x_{l+1}) \tilde{\beta}_{l+1|n}(x_{l+1}) \gamma(x_{l+1}, y_{l+1}) P(x_l, dx_{l+1}) \pi_l(dx_l) \end{aligned}$$



*Dimostrazione.* Procediamo per induzione all'indietro su  $l$ .

i) Passo base:  $l=n-1$ .

$$\begin{aligned}
\pi_{n-1,n|n}(A \times B) &= \\
&= \frac{\int I_A(x_{n-1})I_B(x_n)\beta_{n|n}(x_n)\gamma(x_n, y_n)P(x_{n-1}, dx_n)\sigma_{n-1}(dx_{n-1})}{\int \beta_{n|n}(x_n)\sigma_n(dx_n)} \\
&= \frac{\int I_A(x_{n-1})I_B(x_n)\gamma(x_n, y_n)P(x_{n-1}, dx_n)\sigma_{n-1}(dx_{n-1})}{\int \sigma_n(dx_n)} \\
&= \frac{\int I_A(x_{n-1})I_B(x_n)\gamma(x_n, y_n)P(x_{n-1}, dx_n)\sigma_{n-1}(dx_{n-1})}{\int \gamma(x_n, y_n)P(x_{n-1}, dx_n)\sigma_{n-1}(dx_{n-1})} \\
&= \frac{\int I_A(x_{n-1})I_B(x_n)\gamma(x_n, y_n)P(x_{n-1}, dx_n)\pi_{n-1}(dx_{n-1})}{\int \gamma(x_n, y_n)P(x_{n-1}, dx_n)\pi_{n-1}(dx_{n-1})} \\
&= \int I_A(x_{n-1})I_B(x_n)\tilde{\beta}_{n|n}(x_n)\gamma(x_n, y_n)P(x_{n-1}, dx_n)\pi_{n-1}(dx_{n-1}).
\end{aligned}$$

ii) Passo induttivo: supponiamo valga per  $l+1$  e dimostriamolo per  $l$ .

$$\begin{aligned}
\pi_{l-1,l|n}(A \times B) &= \\
&= \frac{\int I_A(x_l)I_B(x_{l+1})\beta_{l+1|n}(x_{l+1})\gamma(x_{l+1}, y_{l+1})P(x_l, dx_{l+1})\sigma_l(dx_l)}{\int \beta_{l+1|n}(x_{l+1})\sigma_{l+1}(dx_{l+1})} \\
&= \frac{\int I_A(x_l)I_B(x_{l+1})\beta_{l+1|n}(x_{l+1})\gamma(x_{l+1}, y_{l+1})P(x_l, dx_{l+1})\sigma_l(dx_l)}{\int \beta_{l+1|n}(x_{l+1})\gamma(x_{l+1}, y_{l+1})P(x_l, dx_{l+1})\sigma_l(dx_l)} \\
&= \frac{\int I_A(x_l)I_B(x_{l+1})\beta_{l+1|n}(x_{l+1})\gamma(x_{l+1}, y_{l+1})P(x_l, dx_{l+1})\pi_l(dx_l)}{\int \beta_{l+1|n}(x_{l+1})\gamma(x_{l+1}, y_{l+1})P(x_l, dx_{l+1})\pi_l(dx_l)}
\end{aligned}$$

Dobbiamo quindi mostrare che:

$$\frac{\beta_{l+1|n}(x_{l+1})}{\int \beta_{l+1|n}(x_{l+1})\gamma(x_{l+1}, y_{l+1})P(x_l, dx_{l+1})\pi_l(dx_l)} = \tilde{\beta}_{l+1|n}(x_{l+1})$$

Per definizione il membro di destra è:

$$\tilde{\beta}_{l+1|n}(x_{l+1}) = \frac{\int \tilde{\beta}_{l+2|n}(x_{l+2})\gamma(x_{l+2}, y_{l+2})P(x_{l+1}, dx_{l+2})}{\int \gamma(x_{l+1}, y_{l+1})P(x_l, dx_{l+1})\pi_l(dx_l)}$$

Usiamo l'ipotesi induttiva:

$$\begin{aligned}
\tilde{\beta}_{l+1|n}(x_{l+1}) &= \\
&= \frac{\int \beta_{l+2|n}(x_{l+2})\gamma(x_{l+2}, y_{l+2})P(x_{l+1}, dx_{l+2})}{\int \beta_{l+2|n}(x_{l+2})\gamma(x_{l+2}, y_{l+2})P(x_{l+1}, dx_{l+2})\pi_{l+1}(dx_{l+1})} \\
&= \frac{\int \beta_{l+2|n}(x_{l+2})\gamma(x_{l+2}, y_{l+2})P(x_{l+1}, dx_{l+2})}{\int \beta_{l+2|n}(x_{l+2})\gamma(x_{l+2}, y_{l+2})P(x_{l+1}, dx_{l+2})\gamma(x_{l+1}, y_{l+1})P(x_l, dx_{l+1})\pi_l(dx_l)} \\
&= \frac{\beta_{l+1|n}(x_{l+1})}{\int \beta_{l+2|n}(x_{l+2})\gamma(x_{l+2}, y_{l+2})P(x_{l+1}, dx_{l+2})\gamma(x_{l+1}, y_{l+1})P(x_l, dx_{l+1})\pi_l(dx_l)} \\
&= \frac{\beta_{l+1|n}(x_{l+1})}{\int \beta_{l+1|n}(x_{l+1})\gamma(x_{l+1}, y_{l+1})P(x_l, dx_{l+1})\pi_l(dx_l)}
\end{aligned}$$

□

Vediamo infine il problema della previsione.

Iniziamo con una definizione ed un'osservazione circa i processi di Markov omogenei che ci torneranno utili nella dimostrazione del teorema.

Data una funzione misurabile e limitata  $f : E \rightarrow \mathbb{R}$  definiamo la funzione  $Pf : E \rightarrow \mathbb{R}$  ponendo:

$$Pf(x) = \int f(z)P(x, dz)$$

per ogni  $x \in E$ .

Notiamo che per un processo di Markov omogeneo  $(X_k)_{k \geq 0}$  con nucleo di transizione  $P$  si ha:

$$E[f(X_{k+1})|X_0, \dots, X_k] = Pf(X_k)$$

Definiamo ora ricorsivamente le funzioni  $P^n f = PP^{n-1}f$  e osserviamo che:

$$\begin{aligned}
E[f(X_{k+n})|X_0, \dots, X_k] &= E[E[f(X_{k+n})|X_0, \dots, X_{k+n-1}]|X_0, \dots, X_k] \\
&= E[Pf(X_{k+n-1})|X_0, \dots, X_k] \\
&= E[E[Pf(X_{k+n-1})|X_0, \dots, X_{k+n-2}]|X_0, \dots, X_k] \\
&= E[P^2 f(X_{k+n-2})|X_0, \dots, X_k] \\
&\dots \\
&= E[P^n f(X_k)|X_0, \dots, X_k] \\
&= P^n f(X_k).
\end{aligned}$$

**Teorema 1.21** (Ricorsione per la previsione). . *Sia  $k > n$ , la distribuzione di previsione può essere calcolata ricorsivamente come:*

$$\pi_{k|n}(y_0, \dots, y_n, A) = \int I_A(x)P(x', dx)\pi_{k-1|n}(y_0, \dots, y_n, dx')$$

Per ogni  $A \in \mathcal{E}$  e per ogni  $y_0, \dots, y_n \in F$ , con condizione iniziale  $\pi_{n|n} = \pi_n$ .

*Dimostrazione.* Dalle proprietà della speranza condizionale sappiamo che:

$$E[f(X_k)|Y_0, \dots, Y_n] = E[E[f(X_k)|X_0, Y_0, \dots, X_n, Y_n]|Y_0, \dots, Y_n]$$

Sfruttando la il fatto che il segnale è un processo di Markov otteniamo:

$$E[f(X_k)|X_0, Y_0, \dots, X_n, Y_n] = P^{n-k} f(X_n)$$

Da cui:

$$E[f(X_k)|Y_0, \dots, Y_n] = E[P^{n-k} f(X_n)|Y_0, \dots, Y_n]$$

Ossia:

$$\int f(x)\pi_{k|n}(y_0, \dots, y_n, dx) = \int P^{n-k} f(x)\pi_n(y_0, \dots, y_n, dx)$$

E ora si conclude per induzione.  $\square$

### 1.3 Algoritmo di Baum-Welch.

Nel caso particolare in cui lo spazio degli stati sia un insieme finito di cardinalità  $d$ ,  $E = \{1, \dots, d\}$  e le osservazioni siano non degeneri, possiamo rappresentare le componenti fondamentali di un modello di Markov nascosto come matrici e vettori nel seguente modo:

Notiamo che una funzione  $f : E \rightarrow \mathbb{R}$  è completamente determinata dal vettore  $\mathbf{f} = (f(1), f(2), \dots, f(d))^* \in \mathbb{R}^d$ . Allo stesso modo posso rappresentare la misura iniziale  $\mu$  e le distribuzioni di filtraggio e smoothing fissate le osservazioni  $y_0, \dots, y_n \in F$ . Rappresentiamo con  $\mathbf{1}$  la funzione costante 1.

Il nucleo di transizione  $P$  sarà rappresentato da una matrice  $\mathbf{P}$  i cui coefficienti sono  $\mathbf{P}_{ij} = P(i, \{j\})$ .

Per quanto riguarda la densità di osservazione definisco, al variare di  $y \in F$ , la matrice diagonale  $(\gamma(y))_{ii} = \gamma(i, y)$ .

Avvalendoci delle formule dimostrate nella sezione precedente e della notazione appena introdotta, studiamo le distribuzioni di filtraggio, smoothing e smoothing bivariato nel caso di segnali su spazio a stati finiti.

Omettiamo, nella notazione vettoriale, la dipendenza dalle osservazioni. Per il filtraggio e lo smoothing bivariato seguiamo il seguente schema: partiamo dalla formula del caso generale per ottenere quella del caso a stati finiti.

Iniziamo dal filtraggio. Vogliamo arrivare a far vedere che vale:

$$\begin{cases} \boldsymbol{\pi}_0 = \frac{\boldsymbol{\gamma}(y_0)\boldsymbol{\mu}}{\mathbf{1}^* \boldsymbol{\gamma}(y_0)\boldsymbol{\mu}} \\ \boldsymbol{\pi}_k = \frac{\boldsymbol{\gamma}(y_k)\mathbf{P}^* \boldsymbol{\pi}_{k-1}}{\mathbf{1}^* \boldsymbol{\gamma}(y_k)\mathbf{P}^* \boldsymbol{\pi}_{k-1}} \end{cases}$$

Partiamo dalla definizione ricorsiva della distribuzione di filtraggio non normalizzata. Abbiamo visto nel capitolo precedente che è la seguente:

$$\begin{cases} \sigma_0(y_0, A) = \int_A \gamma(x, y_0)\mu(dx) \\ \sigma_k(y_0, \dots, y_k, A) = \int_A \gamma(x, y_k)P(x', dx)\sigma_{k-1}(y_0, \dots, y_{k-1}, dx') \end{cases}$$

Prendiamo ora  $A=\{m\}$  con  $m \in E$ .

La prima equazione diventa:  $\sigma_0(y_0, \{m\}) = \gamma(m, y_0)\mu(\{m\})$ .

Da cui:

$$\boldsymbol{\sigma}_0 = \boldsymbol{\gamma}(y_0)\boldsymbol{\mu}$$

La seconda equazione diventa:

$$\begin{aligned} \sigma_k(y_0, \dots, y_k, \{m\}) &= \gamma(m, y_k) \int_E P(x', \{m\}) \sigma_{k-1}(y_0, \dots, y_{k-1}, dx') \\ &= \gamma(m, y_k) (\mathbf{P}^m)^* \boldsymbol{\sigma}_{k-1} \end{aligned}$$

Dove con  $P^m$  si indica la colonna  $m$ -esima della matrice di transizione.

Da cui:

$$\boldsymbol{\sigma}_k = \boldsymbol{\gamma}(y_k) \mathbf{P}^* \boldsymbol{\sigma}_{k-1}$$

Osserviamo ora che la distribuzione di filtraggio normalizzata si calcola come

$$\boldsymbol{\pi}_k = \frac{\boldsymbol{\sigma}_k}{\mathbf{1}^* \boldsymbol{\sigma}_k}$$

Da cui:

$$\begin{aligned} \boldsymbol{\pi}_0 &= \frac{\boldsymbol{\gamma}(y_0)\boldsymbol{\mu}}{\mathbf{1}^* \boldsymbol{\gamma}(y_0)\boldsymbol{\mu}} \\ \boldsymbol{\pi}_k &= \frac{\boldsymbol{\gamma}(y_k) \mathbf{P}^* \boldsymbol{\sigma}_{k-1}}{\mathbf{1}^* \boldsymbol{\gamma}(y_k) \mathbf{P}^* \boldsymbol{\sigma}_{k-1}} \end{aligned}$$

Sostituendo  $\boldsymbol{\sigma}_{k-1} = \mathbf{1}^* \boldsymbol{\sigma}_{k-1} \boldsymbol{\pi}_{k-1}$  si ottiene:

$$\boldsymbol{\pi}_k = \frac{\boldsymbol{\gamma}(y_k) \mathbf{P}^* \boldsymbol{\pi}_{k-1}}{\mathbf{1}^* \boldsymbol{\gamma}(y_k) \mathbf{P}^* \boldsymbol{\pi}_{k-1}}$$

Per quanto riguarda lo smoothing bivariato, rappresentiamo la distribuzione  $\pi_{l,l+1|n}$  come una matrice  $\boldsymbol{\pi}_{l,l+1|n}$  i cui elementi sono definiti come  $(\boldsymbol{\pi}_{l,l+1|n})_{ij} = \pi_{l,l+1|n}(\{i\} \times \{j\})$ .

Notiamo inoltre che si ha:

$$(\boldsymbol{\pi}_{l|n})_i = \sum_{j=1}^d (\boldsymbol{\pi}_{l,l+1|n})_{ij}$$

Da cui:

$$\boldsymbol{\pi}_{l|n} = \boldsymbol{\pi}_{l,l+1|n} \mathbf{1}$$

Dimostriamo che per lo smoothing bivariato vale:

$$\begin{cases} \boldsymbol{\pi}_{l,l+1|n} = \text{diag}(\boldsymbol{\pi}_l) \mathbf{P} \boldsymbol{\gamma}(y_{l+1}) \text{diag}(\tilde{\boldsymbol{\beta}}_{l+1|n}) \\ \boldsymbol{\pi}_{n-1,n|n} = \text{diag}(\boldsymbol{\pi}_{n-1}) \mathbf{P} \boldsymbol{\gamma}(y_n) \text{diag}(\tilde{\boldsymbol{\beta}}_{n|n}) \end{cases}$$

E che si ha:

$$\begin{cases} \tilde{\beta}_{k|n} = \frac{\mathbf{P}\gamma(y_{k+1})\tilde{\beta}_{k+1|n}}{\mathbf{1}^*\gamma(y_k)\mathbf{P}^*\pi_{k-1}} \\ \tilde{\beta}_{n|n} = \frac{1}{\mathbf{1}^*\gamma(y_n)\mathbf{P}^*\pi_{n-1}} \end{cases}$$

Dalla sezione precedente sappiamo che vale:

$$\begin{cases} \tilde{\beta}_{k|n}(x, y_0, \dots, y_n) = \frac{\int \tilde{\beta}_{k+1|n}(x', y_0, \dots, y_n) \gamma(x', y_{k+1}) P(x, dx')}{\int \gamma(x', y_k) P(x, dx') \pi_{k-1}(y_0, \dots, y_{k-1}, dx)} \\ \tilde{\beta}_{n|n}(x, y_0, \dots, y_n) = \frac{1}{\int \gamma(x', y_n) P(x, dx') \pi_{n-1}(y_0, \dots, y_{n-1}, dx)} \end{cases}$$

Notiamo che nel caso finito il denominatore diventa  $\mathbf{1}^*\gamma(y_n)\mathbf{P}^*\pi_{n-1}$ , mentre l'espressione  $\int \tilde{\beta}_{k+1|n}(x', y_0, \dots, y_n) \gamma(x', y_{k+1}) P(x, dx')$  diventa effettivamente  $\mathbf{P}\gamma(y_{k+1})\tilde{\beta}_{k+1|n}$ .

Per quanto riguarda l'espressione dello smoothing bivariato, prendiamo l'espressione corrispondente della sezione precedente:

$$\begin{aligned} \pi_{l,l+1|n}(A \times B) &= \\ &= \int I_A(x_l) I_B(x_{l+1}) \tilde{\beta}_{l+1|n}(x_{l+1}) \gamma(x_{l+1}, y_{l+1}) P(x_l, dx_{l+1}) \pi_l(dx_l). \end{aligned}$$

Siano ora  $A = \{p\}$  e  $B = \{q\}$  con  $p, q \in E$ . L'uguaglianza diventa:

$$\pi_{l,l+1|n}(\{p\} \times \{q\}) = \tilde{\beta}_{l+1|n}(q) \gamma(q, y_{l+1}) P(p, \{q\}) \pi_l(\{p\}).$$

Che effettivamente corrisponde all'espressione:

$$\pi_{l,l+1|n} = \text{diag}(\pi_l) \mathbf{P}\gamma(y_{l+1}) \text{diag}(\tilde{\beta}_{l+1|n}).$$

Il caso a stati finiti ci permette di implementare il seguente algoritmo per il calcolo delle distribuzioni di filtraggio, smoothing e smoothing bivariato a partire dalle matrici di osservazione, misura iniziale, matrice di transizione e vettore delle osservazioni.

### Algoritmo di Baum-Welch.

L'idea è quella di implementare una ricorsione forward per calcolare la distribuzione di filtraggio e poi una backward per calcolare la distribuzione di smoothing bivariato sfruttando la distribuzione di filtraggio calcolata in precedenza. Infine si calcola la distribuzione di smoothing attraverso lo smoothing bivariato come visto sopra. Riportiamo il codice in Matlab.

```
function [pi,smoot,bivsmoot]=BWalgorithm(gamma,mu,P,y)
    d=size(mu,1);
    n=size(y,1)-1;
```

```

c=zeros(1,n+1);
pi=zeros(d,n+1);
PI=zeros(d,n+1);
c(1)=ones(1,d)*gamma(:, :, y(1))*mu;
for i=1:d
    pi(i,1)=gamma(i, :, y(1))*mu/c(1);
end
for k=1:n
    for j=1:d
        PI(j,k+1)=gamma(j, :, y(k+1))*(P')*pi(:,k);
    end
    c(k+1)=ones(1,d)*PI(:,k+1);
    for j=1:d
        pi(j,k+1)=PI(j,k+1)/c(k+1);
    end
end
for i=1:d
    BETA(n+1,i)=1/c(n+1);
end
bivsmoot=ones(d,d,n);
smoot=ones(n,d);

for k=1:n
    for i=1:d
        BETA(n+1-k,i)=P(i, :)*gamma(:, :, y(n-k+2))*BETA(n-k+2, :)' /c(n+1-k);
    end
end
for k=1:n
    for j=1:d
        for i=1:d
            D=diag(pi(:,n-k+1));
            Beta=diag(BETA(n-k+2, :));
            bivsmoot(i, j, n+1-k)=D(i, :)*P*gamma(:, :, y(n-k+2))*Beta(:, j);
        end
    end
end
for k=1:n
    for i=1:d
        smoot(n+1-k,i)=bivsmoot(i, :, n+1-k)*ones(d,1);
    end
end
end
end

```

## 1.4 Algoritmo di Viterbi.

Rimanendo nel caso di spazio dei segnali finito e restringendoci al caso di segnali a valori in  $\mathbb{R}$ , occupiamoci ora del seguente problema: vogliamo stimare una sequenza finita di stati  $x_0, \dots, x_n$  a partire dalla corrispondente sequenza finita di osservazioni  $y_0, \dots, y_n$ .

Vogliamo quindi arrivare ad ottenere delle funzioni di  $Y_0, \dots, Y_n$   $\hat{X}_0, \dots, \hat{X}_n$  che siano una “buona” stima per i segnali.

Decidiamo a tal proposito di massimizzare la quantità  $\mathbf{P}(X_k = \hat{X}_k, \forall k \leq n)$ , ossia la probabilità di avere una ricostruzione esatta di tutta la sequenza.

Diamo ora un lemma che ci servirà in seguito:

**Lemma 1.22** (Funzione massimizzante). *Siano  $H : \mathbb{R} \rightarrow [0, \infty[$  una certa funzione data,  $X$  una v.a.r. e  $Y$  una v.a. a valori in  $(B, \mathcal{B})$ .*

*Supponiamo che esista una funzione misurabile  $g : B \rightarrow \mathbb{R}$  t.c. per ogni  $y \in B'$*

$$g(y) = \operatorname{argmax}_{\hat{x} \in \mathbb{R}} \int H(x - \hat{x}) P_{X|Y}(y, dx)$$

dove  $B' \in \mathcal{B}$  t.c.  $\mathbf{P}(Y \in B') = 1$ . Allora  $g$  massimizza  $E[H(X - g'(Y))]$  al variare di  $g'$  tra le funzioni misurabili.

*Dimostrazione.* Notiamo che per costruzione

$$\int H(x - g(Y)) P_{X|Y}(Y, dx) \geq \int H(x - g'(Y)) P_{X|Y}(Y, dx) \text{ q.o.}$$

per ogni funzione  $g'$  misurabile.

Da cui

$$\begin{aligned} E[H(X - g(Y))] &= E[E[H(X - g(Y))|Y]] \\ &= E\left[\int H(x - g(Y)) P_{X|Y}(Y, dx)\right] \\ &\geq E\left[\int H(x - g'(Y)) P_{X|Y}(Y, dx)\right] \\ &= E[E[H(X - g'(Y))|Y]] = E[H(X - g'(Y))]. \end{aligned}$$

□

Procediamo con il problema sopra.

Osserviamo che massimizzare la probabilità sopra equivale a massimizzare il valore atteso di  $\prod_{i=0}^n I_0(X_k - \hat{X}_k)$ .

Utilizzando il lemma 1.22. otteniamo che le funzioni che stiamo cercando sono proprio quelle che rispettano la seguente condizione:

$$\begin{aligned} (f_0(y_0, \dots, y_n), \dots, f_n(y_0, \dots, y_n)) &= \\ &= \operatorname{argmax}_{(\hat{x}_0, \dots, \hat{x}_n)} \int \prod_{i=0}^n I_0(x_k - \hat{x}_k) P_{X_0, \dots, X_n | Y_0, \dots, Y_n}(y_0, \dots, y_n, dx_0, \dots, dx_n) \end{aligned}$$

Usando la formula di Bayes otteniamo che l'integrale vale

$$\frac{\gamma(\hat{x}_0, y_0) \cdots \gamma(\hat{x}_n, y_n) P(\hat{x}_{n-1}, \{\hat{x}_n\}) \cdots P(\hat{x}_0, \{\hat{x}_1\}) \mu(\{\hat{x}_0\})}{\int \gamma(x_0, y_0) \cdots \gamma(x_n, y_n) P(x_{n-1}, dx_n) \cdots P(x_0, dx_1) \mu(dx_0)}$$

Dal momento che il denominatore non dipende da  $\hat{x}_0, \dots, \hat{x}_n$  ho

$$\begin{aligned} & (f_0(y_0, \dots, y_n), \dots, f_n(y_0, \dots, y_n)) = \\ & = \operatorname{argmax}_{(\hat{x}_0, \dots, \hat{x}_n)} \gamma(\hat{x}_0, y_0) \cdots \gamma(\hat{x}_n, y_n) P(\hat{x}_{n-1}, \{\hat{x}_n\}) \cdots P(\hat{x}_0, \{\hat{x}_1\}) \mu(\{\hat{x}_0\}) \end{aligned}$$

Visto che il logaritmo è una funzione crescente possiamo passare al logaritmo:

$$\begin{aligned} & (f_0(y_0, \dots, y_n), \dots, f_n(y_0, \dots, y_n)) = \\ & = \operatorname{argmax}_{(\hat{x}_0, \dots, \hat{x}_n)} [\log(\mu(\{\hat{x}_0\})\gamma(\hat{x}_0, y_0)) + \sum_{k=1}^n (\log P(\hat{x}_{k-1}, \{\hat{x}_k\}) + \log \gamma(\hat{x}_k, y_k))] \end{aligned}$$

Introduciamo ora le funzioni

$$\begin{aligned} v_l(\hat{x}_l) & = \\ & = \max_{(\hat{x}_0, \dots, \hat{x}_{l-1})} [\log(\mu(\{\hat{x}_0\})\gamma(\hat{x}_0, y_0)) + \sum_{k=1}^l (\log P(\hat{x}_{k-1}, \{\hat{x}_k\}) + \log \gamma(\hat{x}_k, y_k))] \end{aligned}$$

**Teorema 1.23** (Ricorsione di Viterbi.). *Le funzioni  $v_l$  soddisfano la ricorsione:*

$$v_l(\hat{x}_l) = \max_{\hat{x}_{l-1}} [v_{l-1}(\hat{x}_{l-1}) + \log P(\hat{x}_{l-1}, \{\hat{x}_l\})] + \log \gamma(\hat{x}_l, y_l)$$

con la condizione iniziale  $v_0(\hat{x}_0) = \log(\mu(\{\hat{x}_0\})\gamma(\hat{x}_0, y_0))$ .

Inoltre le funzioni stimanti soddisfano la ricorsione backward:

$$f_l = \operatorname{argmax}_{\hat{x}_l} [v_l(\hat{x}_l) + \log P(\hat{x}_l, \{f_{l+1}\})]$$

con la condizione terminale

$$f_n = \operatorname{argmax}_{\hat{x}_n} v_n(\hat{x}_n).$$

*Dimostrazione.* Iniziamo a dimostrare il primo fatto. La condizione iniziale si ha per definizione della funzione  $v_0(\hat{x}_0)$ . Dall'ipotesi induttiva si ottiene:

$$\begin{aligned} v_l(\hat{x}_l) & = \\ & = \max_{\hat{x}_{l-1}} [\max_{(\hat{x}_0, \dots, \hat{x}_{l-2})} [\log(\mu(\{\hat{x}_0\})\gamma(\hat{x}_0, y_0)) + \sum_{k=1}^{l-1} (\log P(\hat{x}_{k-1}, \{\hat{x}_k\}) + \log \gamma(\hat{x}_k, y_k))] \\ & + \log P(\hat{x}_{l-1}, \{\hat{x}_l\}) + \log \gamma(\hat{x}_l, y_l)] \\ & = \max_{\hat{x}_{l-1}} [v_{l-1}(\hat{x}_{l-1}) + \log P(\hat{x}_{l-1}, \{\hat{x}_l\}) + \log \gamma(\hat{x}_l, y_l)] \\ & = \max_{\hat{x}_{l-1}} [v_{l-1}(\hat{x}_{l-1}) + \log P(\hat{x}_{l-1}, \{\hat{x}_l\})] + \log \gamma(\hat{x}_l, y_l). \end{aligned}$$

Il secondo fatto è conseguenza delle definizioni date. □



In seguito un'implementazione Matlab dell'algoritmo di Viterbi.

```
function f=viterbi(mu,gamma,P,y)
    %gamma è dx dx(n+1)
    %mu è dx1
    %P è dx dx
    %y è (n+1)x1
    f=zeros(1,1000);
    d=size(mu,1);
    n=size(y,1)-1;
    v=zeros(n+1,d);
    b=zeros(n+2,d);
    for i=1:d
        v(1,i)=log(mu(i))+log(gamma(i,i,y(1)));
    end
    for k=2:n+1
        for i=1:d
            b(k,i)=argmax(v(k-1,:)+log(P(:,i)'));
            v(k,i)=v(k-1,b(k,i))+log(P(b(k,i),i))+log(gamma(i,i,y(k)));
        end
    end
    f(n+1)=argmax(v(n+1,:));
    for k=1:n
        f(n+1-k)=b(n+1-k+1,f(n+1-k+1));
    end
end
```

## 1.5 Algoritmo EM per l'inferenza statistica.

Vediamo ora come agire nel caso in cui alcuni parametri del modello siano sconosciuti. Supponiamo di avere a disposizione una sequenza di osservazioni  $y_0, \dots, y_n$  e di voler individuare, a partire da questa, i parametri.

Formalizziamo la situazione nel modo seguente:

Introduciamo uno spazio misurabile  $(\Theta, \mathcal{H})$  che chiameremo spazio dei parametri.

Per ogni  $\theta \in \Theta$  consideriamo un nucleo di transizione  $P^\theta$ , un nucleo di osservazione  $\Phi^\theta$  e una misura iniziale  $\mu^\theta$ .

L'idea è quella di stimare un parametro  $\hat{\theta} \in \Theta$  tale che il modello di Markov nascosto corrispondente descriva bene le osservazioni.

Quello che si fa solitamente in questo caso è utilizzare il metodo della massima verosimiglianza: vediamo in cosa consiste e da dove deriva.

Supponiamo di avere uno spazio delle osservazioni finito  $F = \{1, 2, \dots, d'\}$ . La probabilità che il modello definito dal parametro  $\theta$  generi la sequenza di osservazioni  $y_0, \dots, y_n$  è data da  $\mathbf{P}^\theta(Y_0 = y_0, \dots, Y_n = y_n)$ . L'idea è quindi quella di scegliere il parametro  $\hat{\theta} \in \Theta$  che massimizzi tale quantità.

Se  $F$  non è finito quelle probabilità saranno solitamente nulle. Possiamo però considerare le densità delle osservazioni al posto delle probabilità. In questo caso supponiamo che la misura di probabilità  $\mathbf{P}^\theta|Y_0, \dots, Y_n$  sia assolutamente continua rispetto ad una qualche misura di probabilità fissata  $\mathbf{Q}|Y_0, \dots, Y_n$  per ogni  $\theta \in \Theta$ . La stima di massima verosimiglianza è allora definita come

$$\hat{\theta}(y_0, \dots, y_n) = \operatorname{argmax}_{\theta \in \Theta} \frac{d\mathbf{P}^\theta|Y_0, \dots, Y_n}{d\mathbf{Q}|Y_0, \dots, Y_n}(y_0, \dots, y_n)$$

Ci chiediamo ora che aspetto abbia  $\frac{d\mathbf{P}^\theta|Y_0, \dots, Y_n}{d\mathbf{Q}|Y_0, \dots, Y_n}$  nel caso di un modello di Markov nascosto.

Ci restringiamo al caso in cui, per ogni  $\theta \in \Theta$ ,  $\Phi^\theta(x, dy)$  ammetta una densità strettamente positiva  $\gamma^\theta(x, y)$  rispetto ad una misura fissata  $\varphi(dy)$ . Denotiamo inoltre con  $\pi_k^\theta$  e  $\sigma_k^\theta$  le distribuzioni normalizzata e non di filtraggio.

Vale la seguente:

**Proposizione 1.24.** *Definiamo  $\mathbf{Q}|Y_0, \dots, Y_n(dy_0, \dots, dy_n) = \varphi(dy_0) \cdots \varphi(dy_n)$ . Allora:*

$$\begin{aligned} \mathbf{L}_n^\theta &:= \frac{d\mathbf{P}^\theta|Y_0, \dots, Y_n}{d\mathbf{Q}|Y_0, \dots, Y_n}(y_0, \dots, y_n) = \sigma_n^\theta(y_0, \dots, y_n, E) \\ &= \sigma_0^\theta(y_0, E) \prod_{k=1}^n \int \gamma^\theta(x, y_k) P^\theta(x', dx) \pi_{k-1}^\theta(y_0, \dots, y_{k-1}, dx') \end{aligned}$$

*Dimostrazione.* Per ogni  $f$  misurabile e limitata si ha:

$$\begin{aligned} \mathbf{E}^\theta[f(Y_0, \dots, Y_n)] &= \int f(y_0, \dots, y_n) \gamma^\theta(x_0, y_0) \cdots \gamma^\theta(x_n, y_n) \\ &\quad \times P^\theta(x_{n-1}, dx_n) \cdots P^\theta(x_0, dx_1) \mu^\theta(dx_0) \varphi(dy_0) \cdots \varphi(dy_n) \\ &= \int f(y_0, \dots, y_n) \sigma_n^\theta(y_0, \dots, y_n, E) \varphi(dy_0) \cdots \varphi(dy_n) \end{aligned}$$

□

Per calcolare questa stima di massima verosimiglianza si introduce l'algoritmo EM (Expectation-Maximization).

Scegliamo  $\mathbf{Q} = \mathbf{P}^{\theta'}$  per un qualsiasi  $\theta' \in \Theta$  così che possiamo scrivere:

$$\hat{\theta}(y_0, \dots, y_n) = \operatorname{argmax}_{\theta \in \Theta} \log \left( \frac{d\mathbf{P}^\theta|Y_0, \dots, Y_n}{d\mathbf{P}^{\theta'}|Y_0, \dots, Y_n}(y_0, \dots, y_n) \right)$$

Dove abbiamo usato il fatto che il logaritmo è una funzione strettamente crescente.

Ora utilizziamo il fatto che

$$\log \left( \frac{d\mathbf{P}^\theta|Y_0, \dots, Y_n}{d\mathbf{P}^{\theta'}|Y_0, \dots, Y_n} \right) = \log \left( \mathbf{E}^{\theta'} \left[ \frac{d\mathbf{P}^\theta}{d\mathbf{P}^{\theta'}}|Y_0, \dots, Y_n \right] \right)$$

Dato che massimizzare rispetto a  $\theta$  questa quantità crea problemi di computazione, si considera poi la quantità:

$$Q_n(\theta, \theta') = \mathbf{E}^{\theta'} \left[ \log \left( \frac{d\mathbf{P}^\theta}{d\mathbf{P}^{\theta'}} \right) \middle| Y_0, \dots, Y_n \right]$$

Questa scelta è giustificata dal seguente lemma:

**Lemma 1.25** (EM lemma). *Se  $\theta = \operatorname{argmax}_{\theta_0 \in \Theta} Q_n(\theta_0, \theta')$ , allora  $\mathbf{L}_n^\theta \geq \mathbf{L}_n^{\theta'}$ . Ossia la verosimiglianza di  $\theta'$  non può essere strettamente superiore a quella di  $\theta$ .*

*Dimostrazione.* Notiamo che

$$\log \mathbf{L}_n^\theta - \log \mathbf{L}_n^{\theta'} = \log \left( \mathbf{E}^{\theta'} \left[ \frac{d\mathbf{P}^\theta}{d\mathbf{P}^{\theta'}} \middle| Y_0, \dots, Y_n \right] \right) \geq Q_n(\theta, \theta')$$

per la disuguaglianza di Jensen condizionale. Poiché  $Q_n(\theta, \theta) = 0$  abbiamo  $Q_n(\theta, \theta') \geq 0$ .  $\square$

Questo ci garantisce infatti che massimizzare la quantità  $Q_n(\theta, \theta')$ , che non ci fa ottenere la stima di massima verosimiglianza, ci fa comunque ottenere un parametro che comporta una verosimiglianza maggiore.

Per necessità computazionali bisogna fare alcune assunzioni circa la forma del modello. Dobbiamo infatti restringerci al caso in cui esistano un nucleo di transizione fissato  $P$ , una misura di probabilità su  $E$   $\mu$  e una misura di probabilità su  $F$   $\varphi$  rispetto alle quali  $P^\theta$ ,  $\Phi^\theta$  e  $\mu^\theta$  abbiano densità della forma seguente:

$$P^\theta(x, dx') = p^\theta(x, x')P(x, dx') = \exp\left(\sum_{l=1}^{d_p} c_l(\theta)p_l(x, x')\right)P(x, dx')$$

$$\Phi^\theta(x, dy) = \gamma^\theta(x, y)\varphi(dy) = \exp\left(\sum_{l=1}^{d_\gamma} t_l(\theta)\gamma_l(x, y)\right)\varphi(dy)$$

$$\mu^\theta(dx) = u^\theta(x)\mu(dx) = \exp\left(\sum_{l=1}^{d_u} q_l(\theta)\mu_l(x)\right)\mu(x)$$

In questo caso si ottiene:

$$\begin{aligned} \log\left(\frac{d\mathbf{P}^\theta}{d\mathbf{P}^{\theta'}}(x_0, \dots, x_n, y_0, \dots, y_n)\right) &= \sum_{k=0}^n \sum_{l=1}^{d_\gamma} [t_l(\theta) - t_l(\theta')] \gamma_l(x_k, y_k) \\ &+ \sum_{k=1}^n \sum_{l=1}^{d_p} [c_l(\theta) - c_l(\theta')] p_l(x_k, y_k) + \sum_{l=1}^{d_u} [q_l(\theta) - q_l(\theta')] u_l(x_0) \end{aligned}$$

Da cui:

$$\begin{aligned}
Q_n(\theta, \theta') &= \sum_{k=0}^n \sum_{l=1}^{d_\gamma} [t_l(\theta) - t_l(\theta')] \int \gamma_l(x, y_k) \pi_{k|n}^{\theta'}(dx) \\
&+ \sum_{k=1}^n \sum_{l=1}^{d_p} [c_l(\theta) - c_l(\theta')] \int p_l(x, x') \pi_{k-1, k|n}^{\theta'}(dx, dx') \\
&+ \sum_{l=1}^{d_u} [q_l(\theta) - q_l(\theta')] \int u_l(x) \pi_{0|n}^{\theta'}(dx)
\end{aligned}$$

L'idea dell'algoritmo è quella di partire da un certo valore  $\theta'$ , massimizzare  $Q_n(\theta, \theta')$  e ottenere così un nuovo candidato parametro la cui verosimiglianza corrispondente sia maggiore di quella del precedente e iterare questo processo fino a che non si ha convergenza attorno ad un certo valore che sarà un massimo locale della funzione di verosimiglianza. È importante assicurarsi che la convergenza si abbia sul massimo globale, il cui parametro corrispondente è ciò che stiamo cercando. Non studiamo in questa tesi il problema della convergenza ma, da quanto detto, si capisce l'importanza di scegliere un adeguato valore iniziale.

## 1.6 Modelli non omogenei.

Vediamo come estendere i risultati ottenuti al caso di modelli non omogenei, questo può essere utile in alcune applicazioni.

**Definizione 1.26** (Processo di Markov non omogeneo). Un processo stocastico  $(X_k)_{k \geq 0}$  sullo spazio degli stati  $(E, \mathcal{E})$  è detto processo di Markov non omogeneo se esiste, per ogni  $k \geq 0$ , un nucleo di transizione  $P_k : E \times \mathcal{E} \rightarrow [0, 1]$  t.c.

$$\mathbf{P}(X_{k+1} \in A | X_0, \dots, X_k) = P_k(X_k, A)$$

per ogni  $k \geq 0$  e per ogni  $A \in \mathcal{E}$ .

*Osservazione 1.27.* Se per ogni  $k \geq 0$  si ha  $P_k = P$  si ottiene la definizione di Processo di Markov omogeneo.

Diamo, per un modello di Markov non omogeneo, la stessa definizione di misura iniziale vista per modelli omogenei.

Modificando la dimostrazione del Lemma 1.4. tenendo conto del fatto che il nucleo di transizione dipende dal tempo considerato, si ottiene la legge di un processo di Markov non omogeneo.

**Lemma 1.28** (Legge di un processo di Markov non omogeneo). *Sia  $(X_k)_{k \geq 0}$  un processo di Markov non omogeneo su  $(E, \mathcal{E})$  definito come sopra e con misura iniziale  $\mu$ . Per ogni funzione misurabile e limitata  $f : E^{k+1} \rightarrow \mathbb{R}$  si ha:*

$$\mathbf{E}[f(X_0, \dots, X_k)] = \int f(x_0, \dots, x_k) P_{k-1}(x_{k-1}, dx_k) \cdots P_0(x_0, dx_1) \mu(dx_0)$$

Allo stesso modo può essere data la definizione di modello di Markov Nascosto non omogeneo:

**Definizione 1.29** (Modello di Markov nascosto non omogeneo). Un processo stocastico  $(X_k, Y_k)_{k \geq 0}$  a valori in  $(E \times F, \mathcal{E} \otimes \mathcal{F})$  è detto modello di Markov nascosto non omogeneo se:

1. per ogni  $k \geq 0$  esiste un nucleo di transizione  $P_k : E \times \mathcal{E} \rightarrow [0, 1]$  ed esiste un nucleo di transizione  $\Phi_k : E \times \mathcal{F} \rightarrow [0, 1]$  tali che per ogni  $g : E \times F \rightarrow \mathbb{R}$  misurabile e limitata si abbia:

$$\mathbf{E}[g(X_{k+1}, Y_{k+1}) | X_0, Y_0, \dots, X_k, Y_k] = \int_{E \times F} g(x, y) \Phi_k(x, dy) P_k(X_k, dx)$$

2. esiste una misura di probabilità  $\mu$  su  $E$  t.c.

$$\mathbf{E}[g(X_0, Y_0)] = \int_{E \times F} g(x, y) \Phi_0(x, dy) \mu(dx)$$

per ogni  $g : E \times F \rightarrow \mathbb{R}$  misurabile e limitata.

Se ne può quindi ottenere la legge come è stato fatto per i modelli di Markov nascosti omogenei.

Con opportune considerazioni possono essere poi ottenuti i risultati sulle distribuzioni della sezione 2 per un modello di Markov nascosto non omogeneo.

Ai fini pratici basta sostituire ogni occorrenza di  $P(x_k, dx_{k+1})$  con  $P_k(x_k, dx_{k+1})$  nelle formule viste sopra. Si fa lo stesso per il nucleo di osservazione.

## Capitolo 2

# Applicazione ai canali di trasmissione.

Poniamoci ora nel seguente scenario: un messaggio, modellizzato come una successione  $(B_k)_{k \geq 0}$  di bit i.i.d.  $P(B_k = 0) = P(B_k = 1) = \frac{1}{2}$  viene inviato attraverso un canale disturbato che invia un bit in modo corretto con probabilità  $p$ .

Supponiamo venga utilizzato il metodo della ridondanza e che al posto del singolo bit venga inviata una coppia di bit funzione del bit e dei due bit precedenti.

Prendiamo una funzione che crei una coppia  $(H_k^1, H_k^2)$  fatta nel seguente modo:  $H_k^1 = B_k + B_{k-2}$  e  $H_k^2 = H_k^1 + B_{k-1}$  dove le somme sono fatte modulo 2.

Modellizziamo la situazione tramite un modello di Markov nascosto.

Chiamiamo il nostro processo stocastico  $(X_k, Y_k)_{k \geq 0}$  e definiamo:

$X_k = (B_k, B_{k-1}, B_{k-2})$  e  $Y_k = (Y_k^1, Y_k^2)$ , dove le componenti dell'osservazione rappresentano gli output del canale corrispondenti a  $(H_k^1, H_k^2)$  definiti come sopra. Chiamiamo  $F$  lo spazio delle osservazioni  $\{(0, 0), (0, 1), (1, 0), (1, 1)\}$ .

Prendiamo come spazio degli stati del segnale l'insieme delle terne  $E = \{(x_1, x_2, x_3) | x_i \in \{0, 1\}, \forall i = 1, 2, 3\}$  e ordiniamone gli elementi in ordine crescente secondo la notazione binaria secondo la convenzione per la quale lo stato  $(x_1, x_2, x_3)$  corrisponderà alla notazione binaria  $x_1x_2x_3$ .

Dalle proprietà della matrice di transizione  $\mathbf{P}$ , si arriva a trovare:

$$\mathbf{P} = \begin{bmatrix} 0.5 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 \\ 0.5 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0 & 0.5 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0.5 & 0 & 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0.5 \\ 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0.5 \end{bmatrix}$$

Per quanto riguarda la misura iniziale  $\vec{\mu}$  si ha  $\vec{\mu} = \frac{1}{8}\vec{1}$ .

Le matrici di osservazione risultano essere le seguenti al variare dei valori che può assumere  $Y_k$ :

$$\gamma((0, 0)) = 4 \operatorname{diag}([p^2, (1-p)^2, p(1-p), p(1-p), (1-p)^2, p^2, p(1-p), p(1-p)]),$$

$$\gamma((0, 1)) = 4 \operatorname{diag}([p(1-p), p(1-p), p^2, (1-p)^2, p(1-p), p(1-p), (1-p)^2, p^2]),$$

$$\gamma((1, 0)) = 4 \operatorname{diag}([p(1-p), p(1-p), (1-p)^2, p^2, p(1-p), p(1-p), p^2, (1-p)^2]),$$

$$\gamma((1, 1)) = 4 \operatorname{diag}([(1-p)^2, p^2, p(1-p), p(1-p), p^2, (1-p)^2, p(1-p), p(1-p)]).$$

Mentre  $\vec{\varphi} = \frac{1}{4}(1, 1, 1, 1)^*$ .

Mostriamo che questo è in effetti un modello di Markov nascosto e che nucleo di osservazione e transizione corrispondono a quelli esposti sopra.

Supponiamo di aggiungere all'inizio del messaggio due bit  $B_{-2}$  e  $B_{-1}$  costruiti casualmente (secondo la distribuzione di probabilità degli altri).

Prendiamo  $f : E \times F \rightarrow \mathbb{R}$  misurabile e limitata e consideriamo

$$\mathbf{E}[f(X_{k+1}, Y_{k+1}) | X_0, Y_0, \dots, X_k, Y_k]$$

*Osservazione 2.1.*  $\sigma(X_0, Y_0, \dots, X_k, Y_k) = \sigma(B_{-2}, \dots, B_k, Y_0, \dots, Y_k)$

Da cui:

$$\begin{aligned} \mathbf{E}[f(X_{k+1}, Y_{k+1}) | X_0, Y_0, \dots, X_k, Y_k] &= \\ &= \mathbf{E}[f((B_{k+1}, B_k, B_{k-1}), Y_{k+1}) | B_{-2}, \dots, B_k, Y_0, \dots, Y_k] \\ &= \mathbf{E}[f((B_{k+1}, B_k, B_{k-1}), Y_{k+1}) | B_{k-1}, B_k] \\ &= \mathbf{E}[f((B_{k+1}, b_k, b_{k-1}), Y_{k+1})]_{b_{k-1}=B_{k-1}, b_k=B_k} \end{aligned}$$

Dove il primo passaggio è giustificato dall'osservazione sopra, il secondo dal fatto che  $Y_{k+1}$  è funzione solo dei bit  $B_{k+1}, B_k, B_{k-1}$  e del rumore e l'ultimo

passaggio sfrutta il freezing lemma ( $Y_{k+1}$  è visto come funzione di  $b_{k-1}, b_k$ ).

$$\begin{aligned}
\mathbf{E}[f(X_{k+1}, Y_{k+1}) | X_0, Y_0, \dots, X_k, Y_k] &= \\
&= \frac{1}{2} [f((0, b_k, b_{k-1}), (b_{k-1}, b_k + b_{k-1}))p^2 + \\
&+ f((0, b_k, b_{k-1}), (b_{k-1} + 1, b_k + b_{k-1} + 1))p(1-p) + \\
&+ f((0, b_k, b_{k-1}), (b_{k-1}, b_k + b_{k-1} + 1))p(1-p) + \\
&+ f((0, b_k, b_{k-1}), (b_{k-1} + 1, b_k + b_{k-1}))(1-p)^2 + \\
&+ f((1, b_k, b_{k-1}), (b_{k-1} + 1, b_k + b_{k-1} + 1))p^2 + \\
&+ f((1, b_k, b_{k-1}), (b_{k-1}, b_k + b_{k-1}))p(1-p) + \\
&+ f((1, b_k, b_{k-1}), (b_{k-1} + 1, b_k + b_{k-1}))p(1-p) + \\
&+ f((1, b_k, b_{k-1}), (b_{k-1}, b_k + b_{k-1} + 1))(1-p)^2]_{b_{k-1}=B_{k-1}, b_k=B_k} \\
&= \frac{1}{2} [f((0, B_k, B_{k-1}), (B_{k-1}, B_k + B_{k-1}))p^2 + \\
&+ f((0, B_k, B_{k-1}), (B_{k-1} + 1, B_k + B_{k-1} + 1))p(1-p) + \\
&+ f((0, B_k, B_{k-1}), (B_{k-1}, B_k + B_{k-1} + 1))p(1-p) + \\
&+ f((0, B_k, B_{k-1}), (B_{k-1} + 1, B_k + B_{k-1}))(1-p)^2 + \\
&+ f((1, B_k, B_{k-1}), (B_{k-1} + 1, B_k + B_{k-1} + 1))p^2 + \\
&+ f((1, B_k, B_{k-1}), (B_{k-1}, B_k + B_{k-1}))p(1-p) + \\
&+ f((1, B_k, B_{k-1}), (B_{k-1} + 1, B_k + B_{k-1}))p(1-p) + \\
&+ f((1, B_k, B_{k-1}), (B_{k-1}, B_k + B_{k-1} + 1))(1-p)^2]
\end{aligned}$$

Considerando ora la definizione 1.5. e imponendo che nucleo di transizione e osservazione siano quelli descritti sopra, si vede che l'integrale è uguale a quest'espressione.

## 2.1 Impiego dell'algoritmo di Viterbi.

Simuliamo adesso l'invio di un messaggio di 10000 bit, costruito casualmente rispettando le condizioni sopra, e confrontiamo gli errori ottenuti con e senza l'utilizzo dell'algoritmo di Viterbi per la correzione, in funzione del parametro  $p$  affidabilità del canale.

Il codice matlab impiegato è il seguente e la notazione usata è la stessa che abbiamo descritto sopra:

```

%affidabilità canale(ad es. 0.9)
p=0.9;

%costruiamo il messaggio da inviare
N=10000;
B=ones(1,N);
for j=1:N

```



```

        if rand<=0.5
            B(j)=0;
        end
    end
end

%salviamo H matrice Nx2
H=zeros(N,2);

%trasformiamo il messaggio come abbiamo detto
for k=3:N
    H(k,1)=mod(B(k)+B(k-2),2);
    H(k,2)=mod(H(k,1)+B(k-1),2);
end
H(1,1)=mod(B(1)+B(N-1),2);
H(1,2)=mod(H(1,1)+B(N),2);
H(2,1)=mod(B(2)+B(N),2);
H(2,2)=mod(H(2,1)+B(1),2);

%"facciamo passare H nel canale"
Y=H;
for k=1:N
    casual=rand;
    if casual>=p
        Y(k,1)=mod(Y(k,1)+1,2);
    end
    casual=rand;
    if casual>=p
        Y(k,2)=mod(Y(k,2)+1,2);
    end
end

%ricostruiamo il messaggio originario in M
M=zeros(1,N);
M(1)=mod(Y(2,2)-Y(2,1),2);
M(1000)=mod(Y(1,2)-Y(1,1),2);
for k=2:N-1
    M(k)=mod(Y(k+1,2)-Y(k+1,1),2);
end

%calcoliamo l'errore di trasmissione
E=mod(B+M,2);
errore=sum(E)

%vediamo invece cosa succede ricostruendo il messaggio
%originario con Viterbi

```

```

mu=1/8.*ones(8,1);

P=[ 0.5 0 0 0 0.5 0 0 0 ;
    0.5 0 0 0 0.5 0 0 0 ;
    0 0.5 0 0 0 0.5 0 0 ;
    0 0.5 0 0 0 0.5 0 0 ;
    0 0 0.5 0 0 0 0.5 0 ;
    0 0 0.5 0 0 0 0.5 0 ;
    0 0 0 0.5 0 0 0 0.5 ;
    0 0 0 0.5 0 0 0 0.5 ;
    ];

%diamo le seguenti definizioni per comodità
a=p*p;
b=(1-p)*(1-p);
c=p*(1-p);

gamma(:,:,1)=4.*diag([a b c c b a c c]);
gamma(:,:,2)=4.*diag([c c a b c c b a]);
gamma(:,:,3)=4.*diag([c c b a c c a b]);
gamma(:,:,4)=4.*diag([b a c c a b c c]);

%definiamo il vettore delle osservazioni
y=zeros(N,1);
for k=1:N
    if Y(k,1)==0
        if Y(k,2)==0
            y(k)=1;
        end
        if Y(k,2)==1
            y(k)=2;
        end
    end
    if Y(k,1)==1
        if Y(k,2)==0
            y(k)=3;
        end
        if Y(k,2)==1
            y(k)=4;
        end
    end
end

%usiamo Viterbi per ottenere delle funzioni che
%approssimino il messaggio
f=viterbi(mu,gamma,P,y);

```

```
%matrice terne
V=zeros(3,N);

for k=1:N
    if f(k)==1
        V(1,k)=0;
        V(2,k)=0;
        V(3,k)=0;
    end
    if f(k)==2
        V(1,k)=0;
        V(2,k)=0;
        V(3,k)=1;
    end
    if f(k)==3
        V(1,k)=0;
        V(2,k)=1;
        V(3,k)=0;
    end
    if f(k)==4
        V(1,k)=0;
        V(2,k)=1;
        V(3,k)=1;
    end
    if f(k)==5
        V(1,k)=1;
        V(2,k)=0;
        V(3,k)=0;
    end
    if f(k)==6
        V(1,k)=1;
        V(2,k)=0;
        V(3,k)=1;
    end
    if f(k)==7
        V(1,k)=1;
        V(2,k)=1;
        V(3,k)=0;
    end
    if f(k)==8
        V(1,k)=1;
        V(2,k)=1;
        V(3,k)=1;
    end
end
end
```

```

Vit=V(1,:);

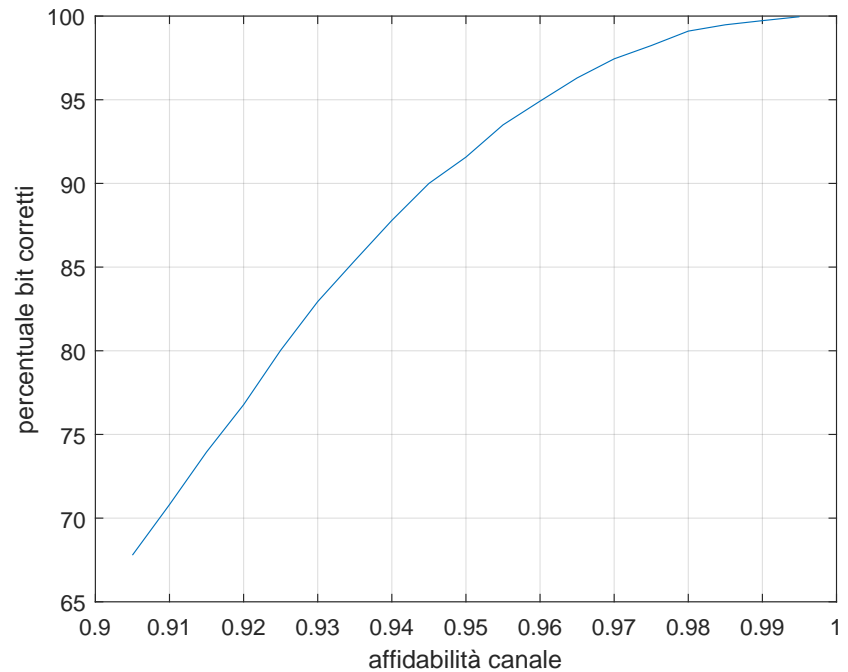
%calcoliamo l'errore
Evit=mod(Vit+B,2);
vit=sum(Evit)

```

*Osservazione 2.2.* Nel codice è stata richiamata la funzione viterbi strutturata come spiegato nella sezione 4 del capitolo 1.

Per la decodifica è stato sfruttato il fatto che  $B_k = H_{k+1}^2 + H_{k+1}^1$  dove la somma è fatta modulo 2. Inoltre, per i primi due segnali, abbiamo supposto il messaggio di struttura circolare.

Con un classico codice di plot, è poi stato costruito il seguente grafico:



Quello che possiamo notare è che viterbi funziona meglio al crescere dell'affidabilità del canale.

## 2.2 Impiego dell'algoritmo EM.

Vogliamo ora utilizzare questo modello per mostrare come si possa fare dell'inferenza statistica utilizzando l'algoritmo EM nel caso particolare in cui sia lo spazio degli stati che quello delle osservazioni siano finiti.

Simuliamo l'invio di un messaggio di 10000 bit nel canale descritto sopra, nel caso in cui l'affidabilità sia  $p = 0.9$ , e immaginiamo di conoscere soltanto la struttura del modello, che è di questo tipo:

La matrice di transizione ha questa struttura:

$$\mathbf{P} = \begin{bmatrix} q & 0 & 0 & 0 & 1-q & 0 & 0 & 0 \\ q & 0 & 0 & 0 & 1-q & 0 & 0 & 0 \\ 0 & q & 0 & 0 & 0 & 1-q & 0 & 0 \\ 0 & q & 0 & 0 & 0 & 1-q & 0 & 0 \\ 0 & 0 & q & 0 & 0 & 0 & 1-q & 0 \\ 0 & 0 & q & 0 & 0 & 0 & 1-q & 0 \\ 0 & 0 & 0 & q & 0 & 0 & 0 & 1-q \\ 0 & 0 & 0 & q & 0 & 0 & 0 & 1-q \end{bmatrix}$$

La misura iniziale è  $\vec{\mu} = \frac{1}{8}\vec{1}$ .

Le matrici di osservazione hanno la struttura esposta precedentemente, dove ora  $p$  è un parametro variabile.

Supponiamo di avere accesso al messaggio ricevuto e di voler calcolare il valore dei parametri  $p$  e  $q$  probabilità che venga trasmesso il bit 0.

Scriviamo allora un codice matlab di questo tipo:

```
%script principale
theta(1,1)=0.3; theta(1,2)=0.8;
theta(2,1)=0.35; theta(2,2)=0.85;
r=2;

%costruiamo il messaggio da inviare
N=10000;
B=ones(1,N);
bit0=0.5;
for j=1:N
    if rand<=bit0
        B(j)=0;
    end
end
%salviamo H matrice Nx2
H=zeros(N,2);
%trasformiamo m come da testo
```

```

for k=3:N
    H(k,1)=mod(B(k)+B(k-2),2);
    H(k,2)=mod(H(k,1)+B(k-1),2);
end
H(1,1)=mod(B(1)+B(N-1),2);
H(1,2)=mod(H(1,1)+B(N),2);
H(2,1)=mod(B(2)+B(N),2);
H(2,2)=mod(H(2,1)+B(1),2);
%"facciamo passare H nel canale"
Y=H;
prob=0.9;
for k=1:N
    casual=rand;
    if casual>=prob
        Y(k,1)=mod(Y(k,1)+1,2);
    end
    casual=rand;
    if casual>=prob
        Y(k,2)=mod(Y(k,2)+1,2);
    end
end

%definisco il vettore delle osservazioni
y=zeros(N,1);
for k=1:N
    if Y(k,1)==0
        if Y(k,2)==0
            y(k)=1;
        end
        if Y(k,2)==1
            y(k)=2;
        end
    end
    if Y(k,1)==1
        if Y(k,2)==0
            y(k)=3;
        end
        if Y(k,2)==1
            y(k)=4;
        end
    end
end
end

while abs(theta(r,1)-theta(r-1,1))>0.0001 || abs(theta(r,2)-theta(r-1,2))>0.0001
    q=theta(r,1);
    p=theta(r,2);

```

```

%questa parte dello script salva la misura iniziale
%e nucleo di transizione e osservazione

mu=1/8.*ones(8,1);

P=[ q 0 0 0 1-q 0 0 0 ;
    q 0 0 0 1-q 0 0 0 ;
    0 q 0 0 0 1-q 0 0 ;
    0 q 0 0 0 1-q 0 0 ;
    0 0 q 0 0 0 1-q 0 ;
    0 0 q 0 0 0 1-q 0 ;
    0 0 0 q 0 0 0 1-q ;
    0 0 0 q 0 0 0 1-q ;
    ];

%diamo le seguenti definizioni per comodità
a=p*p;
b=(1-p)*(1-p);
c=p*(1-p);

gamma(:,:,1)=4.*diag([a b c c b a c c]);
gamma(:,:,2)=4.*diag([c c a b c c b a]);
gamma(:,:,3)=4.*diag([c c b a c c a b]);
gamma(:,:,4)=4.*diag([b a c c a b c c]);

[pi,smoot,bivsmoot]=BWalgorithm(gamma,mu,P,y);
num=sum(bivsmoot(1,1,:))+sum(bivsmoot(2,1,:))+sum(bivsmoot(3,2,:))
+sum(bivsmoot(4,2,:))+sum(bivsmoot(5,3,:))+sum(bivsmoot(6,3,:))
+sum(bivsmoot(7,4,:))+sum(bivsmoot(8,4,:));
den=num+(sum(bivsmoot(1,5,:))+sum(bivsmoot(2,5,:))+sum(bivsmoot(3,6,:))
+sum(bivsmoot(4,6,:))+sum(bivsmoot(5,7,:))+sum(bivsmoot(6,7,:))
+sum(bivsmoot(7,8,:))+sum(bivsmoot(8,8,:)));
theta(r+1,1)=num/den

%calcolo dati per theta(r+1,2)
C=0;
A=0;
B=0;
for k=1:N-1
    if y(k)==1
        %C=C+smoot(k,3)+smoot(k,4)+smoot(k,7)+smoot(k,8);
        A=A+smoot(k,1)+smoot(k,6);
        B=B+smoot(k,2)+smoot(k,5);
    end
    if y(k)==2

```

```

        %C=C+smoot(k,1)+smoot(k,2)+smoot(k,5)+smoot(k,6);
        A=A+smoot(k,3)+smoot(k,8);
        B=B+smoot(k,4)+smoot(k,7);
    end
    if y(k)==3
        %C=C+smoot(k,1)+smoot(k,2)+smoot(k,5)+smoot(k,6);
        A=A+smoot(k,4)+smoot(k,7);
        B=B+smoot(k,3)+smoot(k,8);
    end
    if y(k)==4
        %C=C+smoot(k,3)+smoot(k,4)+smoot(k,7)+smoot(k,8);
        A=A+smoot(k,2)+smoot(k,5);
        B=B+smoot(k,1)+smoot(k,6);
    end
end

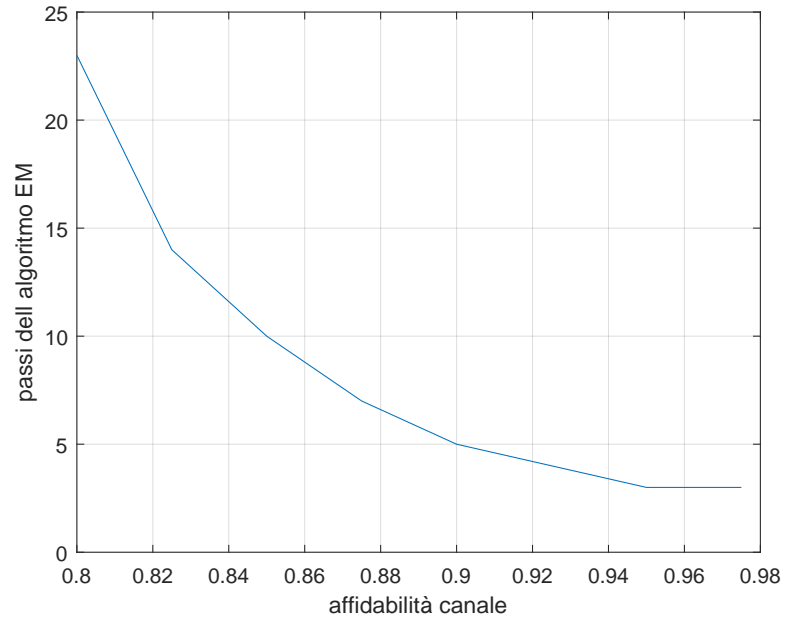
theta(r+1,2)=1/2 + (A-B)/(2*N)
r=r+1
end

```

Nello script è richiamata la funzione BWalgorithm, implementazione dell'algoritmo di Baum-Welch, che calcola le distribuzioni di filtraggio, smoothing e smoothing bivariato come spiegato nella sezione 3 del capitolo precedente.

Con uno script di plot si è costruito poi il seguente grafico per studiare i tempi di convergenza dell'algoritmo EM in funzione dell'affidabilità del canale. È stato tenuto il parametro relativo alla probabilità che un bit del messaggio sia 0 fissa ad  $\frac{1}{2}$  ed è stato posto come criterio di convergenza che la differenza tra due parametri successivi fosse minore di 0.0001. Facendo partire il parametro relativo all'affidabilità del canale da 0.05 in meno del valore reale si è ottenuto questo:





Quello che possiamo notare è che i passi necessari all'algoritmo EM per convergere diminuiscono all'avvicinarsi dell'affidabilità del canale ad 1.

## Capitolo 3

# Un modello di tipo Galton-Watson.

### 3.1 Modellizzazione.

Utilizziamo ora i modelli di Markov nascosti per modellizzare la seguente situazione.

C'è un virus che si diffonde tra la popolazione. Una persona infetta dal virus infetta solo nell'unità di tempo successiva al contagio. Ogni persona infetta infetta i sani seguendo una Poisson di parametro  $\mu$  se non è ancora stata diagnosticata e  $\lambda$  se è stata diagnosticata (questo grazie a delle opportune misure di contenimento del contagio). La probabilità con la quale una persona infetta viene diagnosticata, ossia la potenza del test, è  $\pi$  e il test viene eseguito ogni volta su tutta la popolazione.

Procediamo con la modellizzazione della situazione.

Chiamiamo  $X_n$  la variabile che rappresenta i nuovi contagiati al tempo  $n$ ,  $D_n$  i nuovi diagnosticati al tempo  $n$  e  $Z_n$  i nuovi infetti non diagnosticati. Allora si ha:

$$X_{n+1} = X_{n+1}^D + X_{n+1}^Z \text{ dove}$$

- $X_{n+1}^D = \sum_{i=1}^{D_n} Q_{n,i}$  con  $(Q_{n,i})_{n,i \geq 1}$  i.i.d. Poisson( $\lambda$ )
- $X_{n+1}^Z = \sum_{i=1}^{Z_n} U_{n,i}$  con  $(U_{n,i})_{n,i \geq 1}$  i.i.d. Poisson( $\mu$ )

$$D_{n+1} = \sum_{i=1}^{X_{n+1}} B_{n,i} \text{ con } (B_{n,i})_{n,i \geq 1} \text{ i.i.d. Bernoulli}(\pi)$$

e  $Z_{n+1} = X_{n+1} - D_{n+1}$ .

Vale inoltre la seguente proposizione.

**Proposizione 3.1.** *Il processo stocastico  $((D_n, Z_n), D_n)_{n \geq 1}$  definito su  $((\mathbb{N} \times \mathbb{N}, \mathbb{N}), (\mathcal{P}(\mathbb{N}) \times \mathcal{P}(\mathbb{N}), \mathcal{P}(\mathbb{N})))$  è un modello di Markov nascosto. In particolare  $(D_n, Z_n)_{n \geq 1}$  è Markov omogeneo con operatore di transizione  $N((r, s), \{p\} \times \{q\}) = e^{-(r\lambda + s\mu)} (r\lambda + s\mu)^{(q+p)} \pi^p (1 - \pi)^q \frac{1}{p!q!}$*

*Dimostrazione.* Iniziamo mostrando che  $(D_n, Z_n)_{n \geq 1}$  è Markov omogeneo. Dobbiamo far vedere che esiste  $N : (\mathbb{N} \times \mathbb{N}) \times (\mathcal{P}(\mathbb{N}) \otimes \mathcal{P}(\mathbb{N})) \rightarrow [0, 1]$  t.c

$$\mathbf{P}((D_{n+1}, Z_{n+1}) \in A \times B | D_0, Z_0, \dots, D_n, Z_n) = N((D_n, Z_n), A \times B)$$

Iniziamo sviluppando il membro di sinistra.

$$\begin{aligned} \mathbf{P}((D_{n+1}, Z_{n+1}) \in A \times B | D_0, Z_0, \dots, D_n, Z_n) &= \\ &= E[I_{A \times B}(D_{n+1}, Z_{n+1}) | D_0, Z_0, \dots, D_n, Z_n] \\ &= E[I_{A \times B}(\sum_{i=1}^{X_{n+1}} B_{n,i}, X_{n+1} - \sum_{i=1}^{X_{n+1}} B_{n,i}) | D_0, Z_0, \dots, D_n, Z_n] \\ &= E[I_{A \times B}(\sum_{i=1}^{X_{n+1}} B_{n,i}, X_{n+1} - \sum_{i=1}^{X_{n+1}} B_{n,i}) | D_n, Z_n] \end{aligned}$$

Dove l'ultimo passaggio è giustificato dal fatto che  $X_{n+1}$  dipende soltanto da  $D_n$  e  $Z_n$ .

Vale inoltre:

$$\begin{aligned} E[I_{A \times B}(\sum_{i=1}^{X_{n+1}} B_{n,i}, X_{n+1} - \sum_{i=1}^{X_{n+1}} B_{n,i}) | D_n, Z_n] &= \\ E[E[I_{A \times B}(\sum_{i=1}^{X_{n+1}} B_{n,i}, X_{n+1} - \sum_{i=1}^{X_{n+1}} B_{n,i}) | (U_{i,n})_{i \geq 1}, (Q_{i,n})_{i \geq 1}, D_n, Z_n] | D_n, Z_n] \end{aligned}$$

$$\begin{aligned} E[I_{A \times B}(\sum_{i=1}^{X_{n+1}} B_{n,i}, X_{n+1} - \sum_{i=1}^{X_{n+1}} B_{n,i}) | (U_{i,n})_{i \geq 1}, (Q_{i,n})_{i \geq 1}, D_n, Z_n] &= \\ = \sum_{k=0}^{X_{n+1}} I_A(k) I_B(X_{n+1} - k) \pi^k (1 - \pi)^{X_{n+1} - k} \binom{X_{n+1}}{k} \end{aligned}$$

$$\begin{aligned} E[I_{A \times B}(\sum_{i=1}^{X_{n+1}} B_{n,i}, X_{n+1} - \sum_{i=1}^{X_{n+1}} B_{n,i}) | D_n, Z_n] &= \\ = E[\sum_{k=0}^{X_{n+1}} I_A(k) I_B(X_{n+1} - k) \pi^k (1 - \pi)^{X_{n+1} - k} \binom{X_{n+1}}{k} | D_n, Z_n] \\ = \sum_{l=0}^{\infty} e^{-(D_n \lambda + Z_n \mu)} \frac{(D_n \lambda + Z_n \mu)^l}{l!} \sum_{k=0}^l I_A(k) I_B(l - k) \pi^k (1 - \pi)^{l - k} \binom{l}{k} \end{aligned}$$

Calcoliamo ora  $N((r, s), \{p\} \times \{q\})$ .

$$\begin{aligned}
N((r, s), \{p\} \times \{q\}) &= \\
&= \sum_{l=0}^{\infty} e^{-(r\lambda+s\mu)} \frac{(r\lambda+s\mu)^l}{l!} \sum_{k=0}^l I_p(k) I_q(l-k) \pi^k (1-\pi)^{l-k} \binom{l}{k} \\
&= e^{-(r\lambda+s\mu)} \frac{(r\lambda+s\mu)^{(q+p)}}{(q+p)!} \pi^p (1-\pi)^q \binom{q+p}{p} \\
&= e^{-(r\lambda+s\mu)} (r\lambda+s\mu)^{(q+p)} \pi^p (1-\pi)^q \frac{1}{p!q!}
\end{aligned}$$

Mostriamo infine che  $((D_n, Z_n), D_n)_{n \geq 1}$  è modello di Markov nascosto.

Dobbiamo cercare un nucleo di transizione e un nucleo di osservazione per il modello che rispettino le proprietà di un modello di Markov nascosto.

Consideriamo  $g : (\mathbb{N} \times \mathbb{N}) \times \mathbb{N} \rightarrow \mathbb{R}$  misurabile e limitata.

$$\begin{aligned}
E[g((D_{k+1}, Z_{k+1}), D_{k+1} | D_0, Z_0, \dots, D_k, Z_k] &= \\
&= E[f(D_{k+1}, Z_{k+1}) | D_0, Z_0, \dots, D_k, Z_k] \\
&= \int f(z_1, z_2) N((D_k, Z_k), (dz_1, dz_2))
\end{aligned}$$

Dove  $f$  è un'opportuna funzione.

Se considero  $\Phi : (\mathbb{N} \times \mathbb{N}) \times \mathcal{P}(\mathbb{N}) \rightarrow [0, 1]$  che vale 1 quando la prima componente è contenuta nella parte di  $\mathbb{N}$  e 0 altrimenti ottengo un nucleo di osservazione per il modello.  $\square$

*Osservazione 3.2.* Notiamo che il nucleo di osservazione non rispetta le ipotesi di non degenericità (definizione 1.8). Abbiamo quindi ottenuto un modello di Markov nascosto con osservazioni degeneri.

*Osservazione 3.3.* Questo modello è stato studiato in [1] usando metodi Monte-Carlo, applicandolo oltre ai dati simulati anche a quelli raccolti durante le epidemie di SARS (anno 2003).

## 3.2 Alcune simulazioni.

Mantenendo la stessa notazione della sezione precedente, chiamiamo  $d_n = E[D_n]$  e  $z_n = E[Z_n]$ . Vogliamo utilizzare i valori attesi di queste variabili per comprendere l'andamento dell'epidemia per  $n \rightarrow \infty$ . Iniziamo con lo studiare  $E[X_{n+1}]$ .

$$\begin{aligned}
E[X_{n+1}] &= E[X_{n+1}^D] + E[X_{n+1}^Z] \\
&= E[E[\sum_{i=1}^{D_n} Q_{n,i} | D_n]] + E[E[\sum_{i=1}^{Z_n} U_{n,i} | Z_n]] \\
&= E[\lambda D_n] + E[\mu Z_n] \\
&= \lambda E[D_n] + \mu E[Z_n].
\end{aligned}$$

In modo analogo si ottiene

$$d_{n+1} = E[D_{n+1}] = \pi E[X_{n+1}] = \pi(\lambda d_n + \mu z_n)$$

$$z_{n+1} = E[Z_{n+1}] = (1 - \pi)E[X_{n+1}] = (1 - \pi)(\lambda d_n + \mu z_n)$$

Abbiamo così ottenuto il seguente sistema lineare:

$$\begin{pmatrix} d_{n+1} \\ z_{n+1} \end{pmatrix} = \begin{pmatrix} \pi\lambda & \pi\mu \\ (1-\pi)\lambda & (1-\pi)\mu \end{pmatrix} \begin{pmatrix} d_n \\ z_n \end{pmatrix}$$

Notiamo inoltre che si ha

$$\begin{pmatrix} d_{n+1} \\ z_{n+1} \end{pmatrix} = A^n \begin{pmatrix} d_0 \\ z_0 \end{pmatrix}$$

Studiamo gli autovalori del sistema. Il determinante è nullo, dunque ho un autovalore uguale a 0. La traccia, e dunque l'altro autovalore, è  $r_0 = \pi\lambda + (1 - \pi)\mu$ .

La matrice è  $2 \times 2$  e ho due autovalori distinti, quindi è diagonalizzabile. Consideriamo una base di autovettori. Se ci spostiamo in questa base otteniamo una matrice diagonale il cui unico coefficiente non nullo è  $r_0$ . Osserviamo inoltre che il vettore  $\begin{pmatrix} d_0 \\ z_0 \end{pmatrix}$  non è parallelo alla direzione di un autovettore relativo all'autovalore nullo. Questo è dovuto al fatto che la matrice ha tutti coefficienti positivi e dunque un autovettore relativo all'autovalore 0 ha una componente positiva e una negativa, mentre  $\begin{pmatrix} d_0 \\ z_0 \end{pmatrix}$  ha entrambe le componenti non negative. Quindi se ci spostiamo in una base di autovettori,  $\begin{pmatrix} d_0 \\ z_0 \end{pmatrix}$  ha una componente non nulla in corrispondenza dell'autovalore  $r_0$ .

L'andamento del sistema dipende quindi dal comportamento della potenza  $n$ -esima di  $r_0$  per  $n \rightarrow \infty$ . Se l'autovalore è maggiore di 1 avremo una crescita esponenziale, se è minore di 1 una decrescita dell'epidemia e se è 1 ci troviamo in un caso critico nel quale l'epidemia assume un comportamento irregolare che somiglia a quello di una passeggiata aleatoria.

Simuliamo adesso alcune situazioni e vediamo cosa accade in questi casi.

Supponiamo che inizialmente il virus si diffonda indisturbato tra la popolazione ( $\pi=0$ ) con parametro  $\mu = 2$ . Al tempo  $T=11$  ci si "accorge" della presenza del virus e si introducono un test di potenza  $\pi = 0.7$  e una quarantena strettissima ( $\lambda = 0$ ). Riportiamo in seguito lo script matlab della simulazione e i grafici ottenuti.

```
%gli infetti infettano all'istante successivo
%e poi muoiono
%inizialmente il virus contagia indisturbato
%all'istante T+1 vengono introdotti una
%quarantena strettissima e un test diagnostico
T=10;
%parametro poisson contagio
```

```

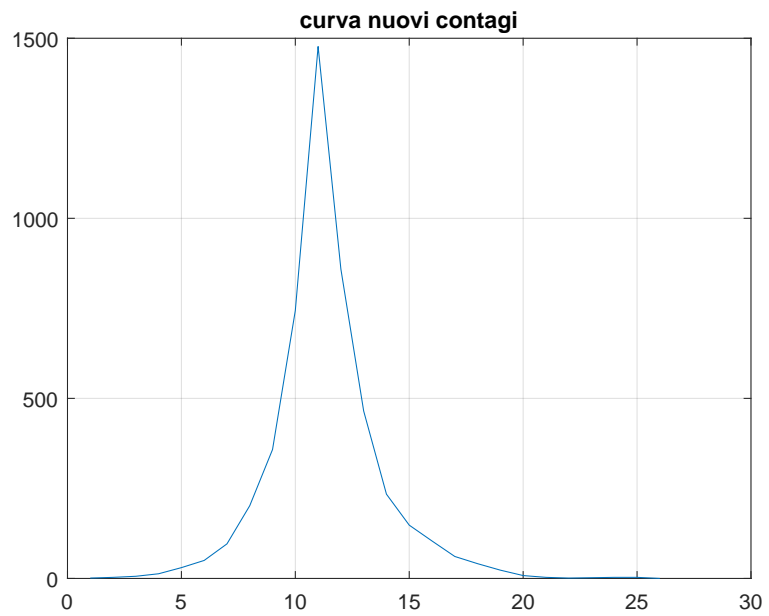
mu=2;
%popolazione
M=100000;
%condizioni iniziali
%infetti liberi iniziali t=1
Z(1)=1;
%diagnosticati t=1
D(1)=0;
%nuovi infetti all'istante t=1
X(1)=Z(1)+D(1);
%infetti totali t=1
Y(1)=X(1);
%contatore istanti
i=1;

while Y(i)<M && X(i)~=0 && i<T
    i=i+1;
    %nuovi infettati all'istante t=i+1
    X(i)=poissrnd(mu*Z(i-1))
    D(i)=0;
    Z(i)=X(i)-D(i);
    Y(i)=Y(i-1)+X(i);
end
if Y(i)>M
    X(i)=X(i)-Y(i)+M;
    Y(i)=Y(i-1)+X(i);
    Z(i)=X(i)-D(i);
end

%introduco il test
%parametro bernoulli diagnosticamento
pi=0.7;
while Y(i)<M && X(i)~=0
    i=i+1;
    %nuovi infettati all'istante t=i+1
    X(i)=poissrnd(mu*Z(i-1))
    D(i)=0;
    for j=1:X(i)
        if rand<pi
            D(i)=D(i)+1;
        end
    end
    Z(i)=X(i)-D(i);
    Y(i)=Y(i-1)+X(i);
end
if Y(i)>M

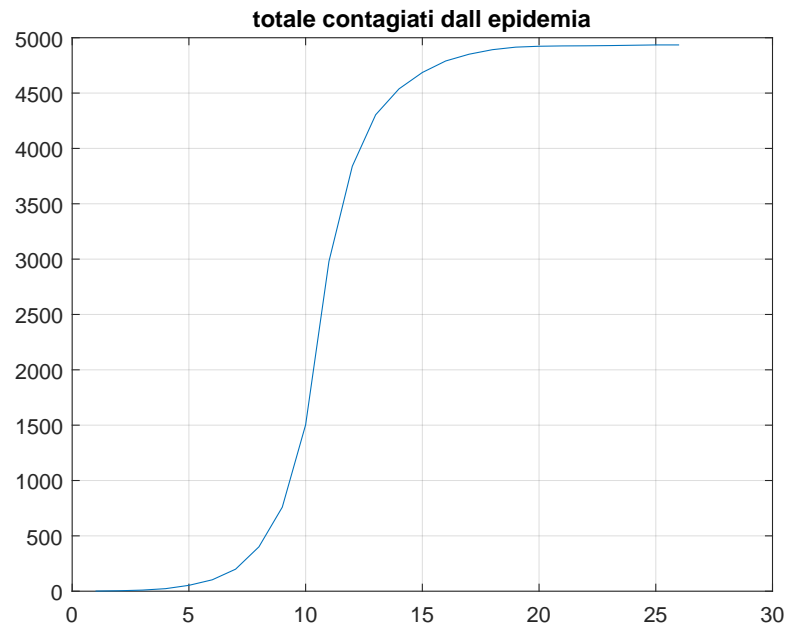
```

```
X(i)=X(i)-Y(i)+M;
D(i)=0;
for j=1:X(i)
    if rand<pi
        D(i)=D(i)+1;
    end
end
Z(i)=X(i)-D(i);
Y(i)=Y(i-1)+X(i);
end
plot(X)
grid on
title('curva nuovi contagi')
figure
plot(Y)
grid on
title('totale contagiati dall epidemia')
```



Quello che si osserva è una curva a campana con picco dei contagi all'istante effettivo dell'introduzione dei test e della quarantena  $T=11$ .





In questo grafico possiamo notare una stabilizzazione dei contagi dopo l'istante 20 che corrisponde al fatto che non ho quasi più nuovi contagiati.

Studiamolo nei casi di nostro interesse:

1.  $\pi = 0$ ,  $\lambda$  qualsiasi e  $\mu > 1$  con condizioni iniziali  $d_0 = 0$ ,  $z_0 > 1$ .
2.  $\pi \in (0, 1)$   $\lambda = 0$  e  $\mu > 1$  con condizioni iniziali generiche.

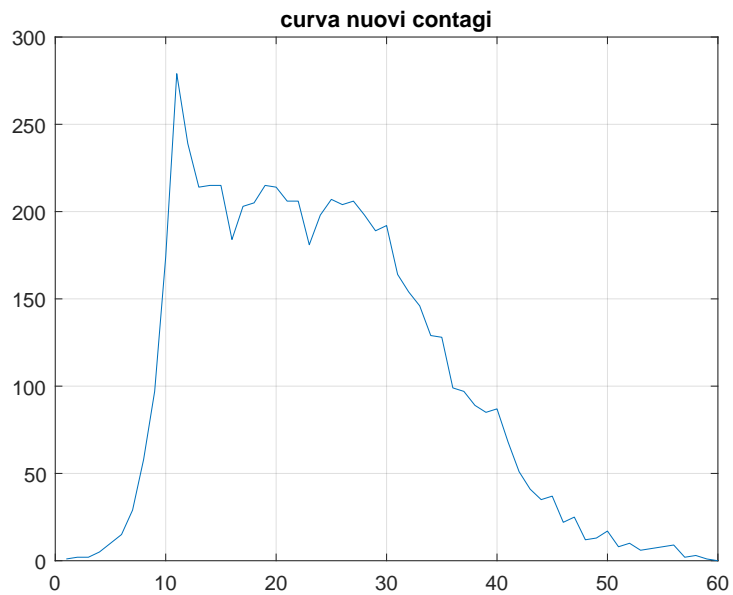
Nel primo caso si ha  $r_0 = \pi\lambda + (1 - \pi)\mu = \mu$  e  $\mu^n \rightarrow \infty$ , da cui la crescita esponenziale.

Dopo il picco abbiamo  $r_0 = \pi\lambda + (1 - \pi)\mu = (1 - \pi)\mu$  in particolare con  $(1 - \pi)\mu = 0.3 \cdot 2 < 1$ , da cui la decrescita.

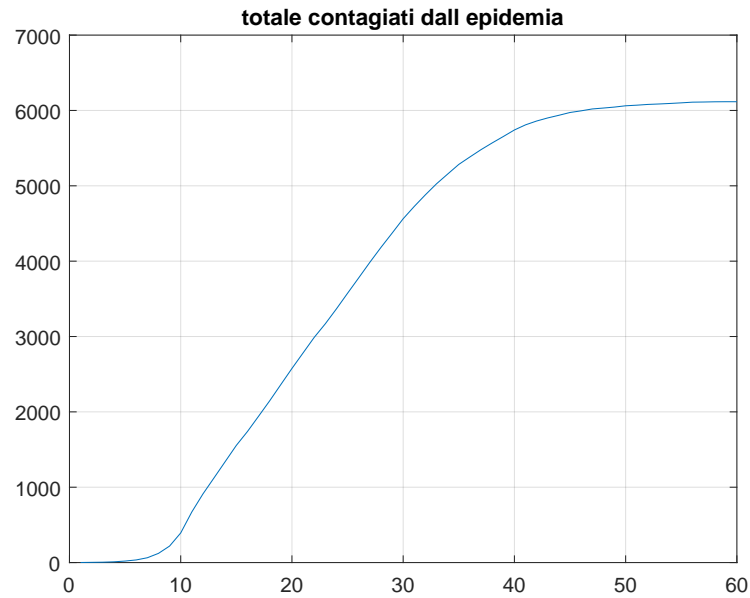
Utilizzando uno script simile al precedente abbiamo poi simulato la seguente situazione:

Il virus si diffonde con parametro di contagio  $\mu = 2$  per gli infetti liberi. Questa volta però anche in un primo momento c'è un test diagnostico, seppur poco potente ( $\pi = 0.2$ ), e una quarantena poco stringente iniziale. Questo comporta che ci sono dei diagnosticati che infettano con un parametro che abbiamo posto  $\lambda = 1$ . All'istante  $T=11$  viene introdotto un test più potente ( $\pi = 0.7$ ) e una quarantena più forte ( $\lambda = 0.5$ ).

Sono risultati i seguenti grafici:



Quello che si può osservare è, come prima, un picco all'istante  $T=11$ , e una discesa "media" successiva costellata da picchi locali.



Essendo in questo caso la discesa più lenta questo si riflette sul grafico dei contagi totali in una crescita che continua in modo "sostanzioso" anche dopo il picco principale.

Studiamo ora questo secondo caso a livello teorico come abbiamo fatto con il precedente.

Nel caso della crescita iniziale si ha  $r_0 = \pi\lambda + (1 - \pi)\mu = 0.2 \cdot 1 + 0.8 \cdot 2 > 1$ . Nel caso della decrescita  $r_0 = \pi\lambda + (1 - \pi)\mu = 0.7 \cdot 0.5 + 0.3 \cdot 2 = 0.95 < 1$ . Il fatto che in questo caso il parametro sia vicino al valore critico spiega il perché del particolare andamento che assume la decrescita.

### 3.3 Algoritmo di Baum-Welch adattato al caso.

Torniamo ora al modello iniziale e vediamo cosa si può fare per utilizzare l'algoritmo di Baum-Welch e ottenere le distribuzioni di filtraggio, smoothing bivariato e smoothing, primo passo per l'inferenza statistica tramite l'algoritmo EM.

Prima di tutto è stato deciso di simulare il modello per ottenere il vettore delle osservazioni, che l'algoritmo di B-W richiede come input.

Questo si può fare con un simulatore come il seguente, molto simile a quello visto in precedenza.

```

%lambda
lambda=1;
%parametro poisson contagio
mu=2;
%parametro bernoulli diagnosticamento
pi=0.7;
%popolazione
M=100;
%contatore già infettati
infetti=2;
%rimuove X, D, Z eventualmente già calcolati
clearvars X D Z
%infetti liberi iniziali t=1
Z(1)=2;
%diagnosticati iniziali t=1
D(1)=0;
%pongo questo per far partire il ciclo
%while nel vettore X salverò i nuovi
%infetti all'istante t
X(1)=Z(1);
%contatore istanti
i=1;

while infetti<M && X(i)~=0
    i=i+1;
    %nuovi infettati all'istante t=i+1
    X(i)=poissrnd(mu*Z(i-1)+lambda*D(i-1))
    D(i)=0;
    for j=1:X(i)
        if rand<pi
            D(i)=D(i)+1;
        end
    end
    end
    Z(i)=X(i)-D(i);
    infetti=infetti+X(i);

```

```

end
if infetti>M
    X(i)=X(i)-infetti+M;
    D(i)=0;
    for j=1:X(i)
        if rand<pi
            D(i)=D(i)+1;
        end
    end
    Z(i)=X(i)-D(i);
end

```

Successivamente è stato scritto uno script come questo:

```

%matrice di transizione e misura iniziale

%misura iniziale: supponiamo inizialmente ci siano
%0 infetti diagnosticati e 1 infetto libero con
%probabilità 1
mis = zeros(M*M, 1);
mis(2) = 1;

%matrice transizione funzione di mu, lambda,pi
%gli stati sono M^2 (D_n,Z_n)
%D_n sta sulle colonne
%N è(M*M,M*M)

N = zeros(M*M, M*M);

for s=1:M
    for r=1:M
        N(r + (s-1)*M, 1) = exp(-((r-1)*lambda+(s-1)*mu));
        for q=2:M
            N(r+(s-1)*M, 1+(q-1)*M)=N(r+(s-1)*M, 1 + (q-2)*M) *
* ((r-1)*lambda+(s-1)*mu)*(1-pi)/(q-1);
        end
        for p=2:M
            N(r + (s-1)*M, p) = N(r+(s-1)*M, p-1)*((r-1)*lambda+(s-1)*mu)*pi/(p-1);
            for q=2:M
                N(r+(s-1)*M, p+(q-1)*M)=N(r+(s-1)*M, p + (q-2)*M) *
* ((r-1)*lambda+(s-1)*mu)*(1-pi)/(q-1);
            end
        end
    end
end
end

```

```

%normalizza le righe di N in modo che la somma sia 1
for i=1:(M*M)
    N(i,:)=N(i,+)/sum(N(i,:));
end

```

Questo script salva la matrice di transizione e la misura iniziale.

Infine trattiamo il fatto che le osservazioni del nostro modello sono degeneri. Questo comporta dei problemi nel momento in cui cerco di implementare l'algoritmo di B-W perché capita che si abbiano delle divisioni per zero. Una soluzione a questo problema potrebbe essere quella di utilizzare del rumore. Questo è stato fatto nel seguente modo: laddove compariva uno 0 nella diagonale delle matrici di osservazione è stato posto un valore  $\varepsilon$  molto piccolo. Per normalizzare la matrice complessiva di osservazione bisognerebbe poi dividere i valori diversi da zero che compaiono sulle righe per la somma dei valori di riga. Questo è stato tralasciato visto che eseguendo delle prove è stato visto che prendendo piccoli valori di epsilon il risultato dall'uno all'altro non cambia.

```
function [filt,smoot,bivsmoot]=BWrumore(mis, N, D, M, epsilon)
```

```

%rumore
epsilon=0.00001;
d=size(mis,1);
n=size(D,2)-1;
c=zeros(1,n+1);

filt=zeros(d,n+1);
bivsmoot=ones(d,d,n);
smoot=ones(n,d);

PI=zeros(d,n+1);
mis2=mis;

%calcolo la distribuzione di filtraggio
for i=1:d
    if mod(i,M)~=D(1)
        mis2(i)=mis(i)*epsilon;
    end
end
c(1)=sum(mis2);
for i=1:d
    if mod(i,M)==D(1)
        filt(i,1)=mis(i)/c(1);
    else
        filt(i,1)=mis(i)*epsilon/c(1);
    end
end
for k=1:n

```

```

    for j=1:d
        if mod(j,M)==D(k+1)
            PI(j,k+1)=N(:,j)'*filt(:,k);
        else
            PI(j,k+1)=N(:,j)'*filt(:,k)*epsilon;
        end
    end
    c(k+1)=ones(1,d)*PI(:,k+1);
    for j=1:d
        filt(j,k+1)=PI(j,k+1)/c(k+1);
    end
end

%calcolo la distribuzione di smoothing biv.
%e di smoothing
for i=1:d
    BETA(n+1,i)=1/c(n+1);
end

for k=1:n
    for i=1:d
        vec=N(i,:);
        for j=1:d
            if mod(j,M)~=D(n-k+2)
                vec(j)=N(i,j).*epsilon;
            end
        end
        BETA(n+1-k,i)=vec*BETA(n-k+2,:)' / c(n+1-k);
    end
end

for k=1:n
    for j=1:d
        for i=1:d
            D=sparse(diag(filt(:,n-k+1)));
            Beta=sparse(diag(BETA(n-k+2,:)));
            vec2=D(i,:)*N;
            for s=1:d
                if mod(s,M)~=D(n-k+2)
                    vec2(s)=vec2(s)*epsilon;
                end
            end
            bivsmoot(i,j,n+1-k)=vec2*Beta(:,j);
        end
    end
end
for k=1:n

```

```

    for i=1:d
        smoot(n+1-k,i)=bivsmoot(i, :,n+1-k)*ones(d,1);
    end
end
end
end

```

La memoria impiegata dalla matrice della distribuzione bivariata è troppo grande per poter considerare un tetto della popolazione adeguato a rappresentare un caso reale. Si conclude affermando che a livello teorico questo algoritmo permetterebbe di ottenere le distribuzioni cercate.

Possiamo però limitarci a calcolare la distribuzione di filtraggio ed utilizzare quella per ottenere, ad un istante di tempo fissato, la distribuzione di probabilità degli infetti nascosti conoscendo la sequenza di infetti diagnosticati dall'istante iniziale a tale istante.

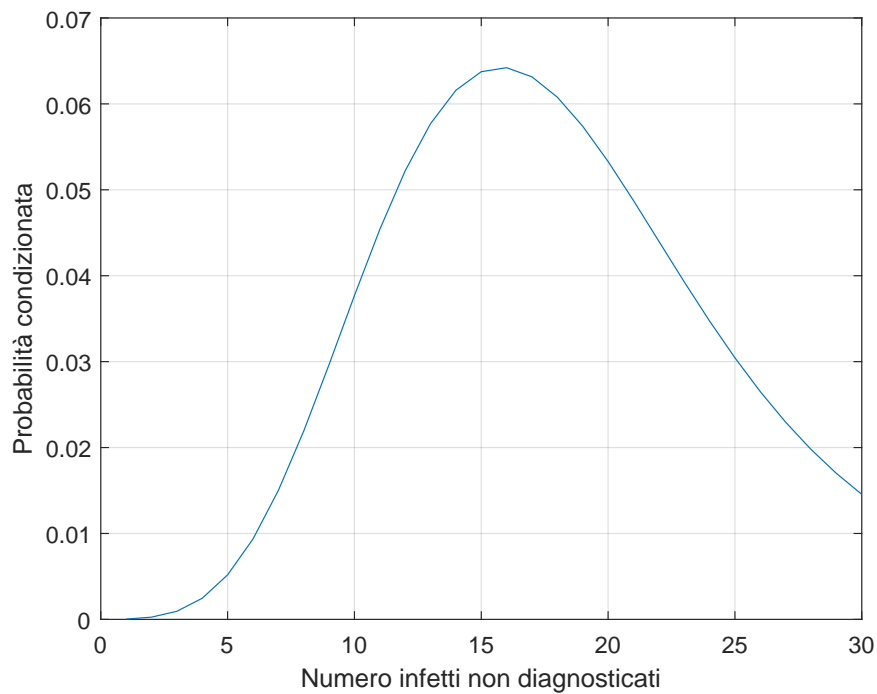
Per fare questo si è fatto andare lo script di simulazione<sup>1</sup> e lo script che salva matrice di transizione e misura iniziale con dei valori dei parametri che permettessero una crescita lenta (nel nostro caso  $\mu = 1.5$ ,  $\lambda = 1$ ,  $\pi = 0.2$ ). Questo per ottenere un buon numero di osservazioni mantenendo un tetto basso della popolazione. Si è poi scelto un istante di tempo (nel nostro caso  $t = 11$ ) e si è sommato sulle componenti del filtraggio in modo adeguato per ottenere la distribuzione condizionata degli infetti nascosti corrispondente a tale istante.

Il vettore delle osservazioni risultante dalla simulazione è  $[0, 0, 0, 0, 0, 1, 1, 0, 2, 1, 5, 6]$  e il numero dei nuovi infetti non diagnosticati all'istante  $t = 11$  è 14. Il grafico ottenuto è il seguente:

---

<sup>1</sup>Nello script è stata modificata la condizione di arresto: il ciclo while si arresta nel caso in cui i nuovi infetti raggiungano il numero M. Non si tiene dunque conto degli infetti precedenti.





Il risultato è consistente con il numero di nuovi infetti non diagnosticati che risultano, all'istante  $t = 11$ , dalla simulazione.

Riportiamo in seguito lo script utilizzato per sommare sulle componenti di filtraggio.

```
vector=zeros(1,M);  
for k=1:M  
    vector(1,k)=sum(filt(1+M*(k-1):+M*k,12));  
end
```

### 3.4 Un'ultima osservazione sull'inferenza.

Dal momento che è la distribuzione di smoothing bivariato a crearci dei problemi, prendendo come riferimento il nucleo di transizione corrispondente a

$\mu = \lambda = 1$  e  $\pi = \frac{1}{2}$  N', vediamo in quali casi l'algoritmo EM non necessita dello smoothing bivariato.

Si ha:

$$\begin{aligned} N((r, s), (p, q)) &= \\ &= \exp(p \cdot \log(1 - \pi) + q \cdot \log(\pi) + (p + q) \cdot \log(\lambda r + \mu s) - r\lambda - s\mu + \dots) N'((r, s), (p, q)) \end{aligned}$$

Dove i puntini rappresentano termini indipendenti da  $\mu$ ,  $\lambda$  e  $\pi$ . Questi termini non ci interessano perché quando vado a derivare l'espressione di  $Q_n$  rispetto ai parametri scompaiono.

Quello che devo fare è "spezzare" il termine  $(p + q) \cdot \log(\lambda r + \mu s)$  in modo che (r,s) non compaia dove compare (p,q) nei termini in cui sono presenti i parametri.

Ponendo  $\lambda = 0$  ottengo  $(p + q) \cdot \log(\mu s) = (p + q) \cdot \log(\mu) + (p + q) \cdot \log(s)$  e ai fini dell'inferenza ci interessa solo il primo addendo, che ha la proprietà richiesta.

Quindi tornando al primo caso simulato, supponendo di sapere che le persone in quarantena non contagiano nessuno ( $\lambda = 0$ ), in linea di principio si potrebbe fare inferenza sulla parte decrescente e ottenere il parametro di contagio  $\mu$  e la potenza del test  $\pi$ . Per fare questo si dovrebbe implementare un algoritmo che calcoli la distribuzione di smoothing senza passare dalla bivariata. Esiste ad esempio l'algoritmo Forward-Backward, che non abbiamo trattato in questa tesi, che calcola la distribuzione di smoothing attraverso quella di filtraggio e che si adatterebbe bene a questo caso.

## Appendice A

# Teorema di Fubini-Tonelli per i nuclei di transizione.

Siano  $(E, \mathcal{E})$  e  $(F, \mathcal{F})$  due spazi misurabili, sia  $\mu$  probabilità su  $(E, \mathcal{E})$  e sia  $N$  nucleo di transizione  $N : E \times \mathcal{F} \rightarrow [0, 1]$ . Il nostro scopo è costruire sullo spazio prodotto  $E \times F$  munito della  $\sigma$ -algebra prodotto  $\mathcal{E} \otimes \mathcal{F}$  una probabilità  $\mathbf{Q}$  tale che si abbia, per ogni rettangolo misurabile  $A \times B$ ,  $\mathbf{Q}(A \times B) = \int_A N(x, B) \mu(dx)$ . Tale probabilità, se esiste, è unica in quanto definita sui rettangoli che generano la  $\sigma$ -algebra prodotto.

**Proposizione A.1.** *Sia  $f : E \times F \rightarrow \mathbb{R}$  misurabile a valori positivi. Allora:*

1. *per ogni  $x \in E$  fissato, la funzione  $y \mapsto f(x, y)$  è  $\mathcal{F}$ -misurabile.*
2. *la funzione  $x \mapsto \int_F f(x, y) N(x, dy)$  è  $\mathcal{E}$ -misurabile.*

*Il primo enunciato si estende poi per differenza ad  $f$  misurabile di segno qualsiasi e il secondo ad  $f$  misurabile e limitata (deve avere senso l'integrale).*

*Dimostrazione.* Se  $f$  è funzione indicatrice di un rettangolo misurabile  $A \times B$  le due proprietà sono di verifica immediata: la preimmagine di un boreliano attraverso la prima funzione mi dà un elemento della  $\sigma$ -algebra generata da  $B$ , che è  $\mathcal{F}$ -misurabile, la seconda funzione diventa  $x \mapsto N(x, B)$  che è  $\mathcal{E}$ -misurabile per le proprietà del nucleo.

Se  $f$  è funzione indicatrice di un insieme misurabile  $C \in \mathcal{E} \otimes \mathcal{F}$  considero la famiglia delle parti di  $\mathcal{E} \otimes \mathcal{F}$  tali che la corrispondente funzione indicatrice rispetta le due proprietà, mostro che questa è una classe monotona stabile per differenza e ottengo dunque che tale famiglia contiene la  $\sigma$ -algebra generata dai rettangoli misurabili (che sono stabili per intersezione).

Quindi per  $f$  semplice i due enunciati sono soddisfatti. Nel caso di  $f$  misurabile positiva considero una successione crescente di funzioni semplici che converge ad  $f$  ed utilizzo il fatto che il limite di funzioni misurabili è misurabile per la prima proprietà e quest'ultimo fatto insieme a Beppo-Levi per la seconda.  $\square$

**Definizione A.2** (Sezione.). Se  $C$  è un sottoinsieme di  $E \times F$  chiamiamo sezione di  $C$  nel punto  $x \in E$  l'insieme  $C_x = \{y \in F \mid (x, y) \in C\}$ .

**Teorema A.3** (Fubini-Tonelli.). *Definiamo, per  $C \in \mathcal{E} \otimes \mathcal{F}$ ,*

$$\mathbf{Q}(C) = \int_E N(x, C_x) \mu(dx)$$

La funzione d'insieme  $\mathbf{Q}$  è una probabilità ed è proprio la probabilità cercata. Inoltre per ogni funzione  $f$  definita su  $E \times F$  misurabile a valori positivi, vale la formula

$$\int \int_{E \times F} f(x, y) d\mathbf{Q}(x, y) = \int_E \left[ \int_F f(x, y) N(x, dy) \right] \mu(dx)$$

*Dimostrazione.* Iniziamo notando che la funzione d'insieme  $C \mapsto \mathbf{Q}(C)$  è  $\sigma$ -additiva per le proprietà del nucleo. Inoltre si ha:

$$\mathbf{Q}(E \times F) = \int_E N(x, F) \mu(dx) = \int_E \mu(dx) = 1.$$

Sui rettangoli misurabili  $A \times B$  si ha:

$$\mathbf{Q}(A \times B) = \int_A N(x, B) \mu(dx)$$

come voluto. Dunque  $\mathbf{Q}$  è la probabilità cercata.

Per quanto riguarda la formula per il calcolo dell'integrale procediamo nel modo usuale.

Se  $f$  è l'indicatrice di un insieme  $C \in \mathcal{E} \otimes \mathcal{F}$  si verifica direttamente che l'uguaglianza è soddisfatta.

Se  $f$  è una funzione semplice sfrutto il fatto che l'integrale di una somma finita è la somma degli integrali e il punto precedente.

Se  $f$  è positiva e misurabile considero una successione crescente di funzioni semplici che la approssima, per ogni funzione della successione l'uguaglianza è soddisfatta per il punto precedente, passo al limite ad entrambi i membri usando Beppo-Levi ed ottengo la formula voluta.  $\square$

# Bibliografia

- [1] K. Glass, N. Becker, M. Clements. Predicting case numbers during infectious disease outbreaks when some cases are undiagnosed. Wiley InterScience, 2006.
- [2] Ramon van Handel. Hidden Markov Models. Lecture Notes, 2008.
- [3] Paolo Baldi. Calcolo delle probabilità. McGraw-Hill, 2007.