

ALMA MATER STUDIORUM · UNIVERSITÀ DI  
BOLOGNA

---

FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E NATURALI  
Corso di Laurea Magistrale in Matematica

**CATENE DI MARKOV  
NASCOSTE  
E GIOCHI STOCASTICI.**

Tesi di Laurea in Calcolo delle Probabilità

Relatore:  
Chiar.mo Prof.  
Massimo Campanino

Presentata da:  
Matteo Umberto Italiani

Sessione III  
Anno Accademico 2013-2014



*alle teste dure come la mia.*



# Indice

<b>Introduzione</b>	<b>iii</b>
<b>1 Catene di Markov omogenee con una infinità numerabile di stati.</b>	<b>1</b>
<b>2 Modelli di Markov Nascosti</b>	<b>5</b>
2.1 Problema della valutazione. . . . .	8
2.1.1 Algoritmo in avanti. . . . .	8
2.1.2 Algoritmo in indietro . . . . .	10
2.1.3 Considerazioni computazionali. . . . .	11
2.2 Problema della decodifica. . . . .	11
2.2.1 Algoritmo di Viterbi. . . . .	12
2.3 Problema dell' apprendimento . . . . .	14
2.3.1 Prova delle formule di ristima. . . . .	17
2.3.2 Formula di ristima con scaling. . . . .	21
2.4 Algoritmi genetici applicati all' addestramento dei parametri. .	24
2.4.1 Descrizione degli algoritmi genetici. . . . .	25
2.4.2 Applicazione al learning problem. . . . .	27
<b>3 Giochi di Markov</b>	<b>29</b>
3.1 Cenni di teoria dei giochi . . . . .	29
3.1.1 Equilibrio di strategie . . . . .	32
3.1.2 Giochi con mancanza di informazioni. . . . .	34

---

<b>4</b>	<b>Applicazione di Modelli di Markov nascosti ai Giochi Markoviani.</b>	<b>37</b>
4.1	Gioco di Renault e strategie ottimali. . . . .	38
4.1.1	Strategie di equilibrio. . . . .	41
4.2	Strategie basate su Modelli di Markov Nascosti. . . . .	43
4.3	Applicazione del metodo contro un giocatore stazionario. . . . .	46
4.4	Prova del metodo contro giocatore ottimale. . . . .	56
4.5	Gioco permanente per lunghi tratti. Una possibile strategia per il giocatore informato. . . . .	61
<b>5</b>	<b>Codici per implementazione in Matlab.</b>	<b>67</b>
5.1	Algoritmi di valutazione. . . . .	67
5.2	Algoritmo di Viterbi. . . . .	69
5.3	Algoritmi di learning dei parametri. . . . .	70
<b>6</b>	<b>Conclusioni e lavori futuri.</b>	<b>75</b>
	<b>Bibliografia</b>	<b>77</b>

# Introduzione

Nell'ambito delle applicazioni assumono notevole importanza i modelli probabilistici, basati sullo studio e sull'utilizzo di variabili casuali. In generale si possono considerare i *processi stocastici*, ovvero modelli in cui le variabili sono sia casuali che mutabili nel tempo, come ad esempio le condizioni atmosferiche su un determinato paese al variare dei giorni di un mese, lo storico delle vincite a giochi di fortuna, i clienti giornalieri che arrivano in un dato negozio.

In questa trattazione si esamineranno dei modelli stocastici in cui il processo non può essere direttamente osservato, ma può essere ricostruito tramite delle osservazioni dipendenti dal processo: ad esempio nel problema del riconoscimento vocale, al posto delle parole vere e proprie si ha a che fare con dei suoni corrispondenti alle parole stesse.

In particolare si supporrà che il processo stocastico alla base del modello sia markoviano, e che gli osservabili in ogni tempo siano dipendenti solo dal valore del processo nello stesso istante.

A tal scopo, nel Capitolo 1 si introdurranno brevemente le *catene di Markov*, che saranno il modello di processo aleatorio preso in esame nel corso di tutta la trattazione.

Successivamente, nel Capitolo 2 verranno presentate le *catene di Markov nascoste*, ovvero i modelli probabilistici sopra introdotti, consistenti nella coppia di processi stocastici  $(X, O)$ , con  $X$  catena di Markov non osservabile direttamente, e  $O$  processo delle osservazioni "scaturite" dalla catena sottostante. Nel Capitolo 2 inoltre verranno risolti i tre problemi classici delle

catene di Markov nascoste: calcolare la probabilità delle sequenze osservate, trovare la sequenza nascosta più probabile in corrispondente a osservazioni date, aggiustare i parametri del modello per spiegare meglio le osservazioni delle quali si è a disposizione.

Le soluzioni di questi problemi, costituiscono dei tool utili al fine di lavorare con schemi di Markov nascosti. L' applicazione presa in esame consiste nel coniugare, per quanto possibile, il concetto di schema di Markov nascosto con la Teoria dei Giochi, in particolare ai giochi stocastici della tipologia introdotta da Renault nella trattazione del 2009 *The value of Markov Chain Game with Lack of Information on One Side* ([11]).

Si introdurranno per questo motivo, nel Capitolo 3, alcuni concetti base della Teoria dei Giochi, mentre nel Capitolo 4 si cercheranno di utilizzare le catene di Markov nascoste per stabilire delle strategie di gioco.

In particolare si confronteranno i risultati ottenuti tramite l' uso di strategie basate sul concetto di catena nascosta con le strategie ottimali, dove è dimostrata l' esistenza di queste ultime, e allo stesso tempo osservare i risultati dell' approccio di Markov nascosto nei casi in cui la ricerca di strategie ottimali è tutt' oggi un problema aperto.



# Capitolo 1

## Catene di Markov omogenee con una infinità numerabile di stati.

In molteplici ambiti scientifici lo studio di fenomeni aleatori, o casuali, ricopre un ruolo di fondamentale importanza.

In particolare si è interessati allo studio di enti casuali che variano nel tempo, come ad esempio il prezzo di mercato di un titolo finanziario, o le condizioni meteo su una determinata città.

Formalmente un fenomeno di questo tipo può essere descritto tramite un processo stocastico  $(X)_{t \in I}$ , ovvero una sequenza di variabili aleatorie (v.a.), distinguendo i casi in cui

- $I$  insieme discreto, ad esempio  $\mathbb{N}$ , in questo caso si parlerà di *processo stocastico a tempo discreto*. In questo caso si considera una successione di variabili aleatorie

$$(X)_n : X_i : \Omega \rightarrow S, \quad i = 1, 2, 3, \dots \quad (1.1)$$

Esempi di processi aleatori a tempo discreto possono essere i risultati di un esperimento ripetuto più volte (in questo caso  $n$  è il numero dell'

## 2.1. Catene di Markov omogenee con una infinità numerabile di stati.

esperimento), o il prezzo di un titolo finanziario alla chiusura della giornata di mercato.

- $I$  insieme continuo, ad esempio  $\mathbb{R}$ :

$$(X)_t : \mathbb{R} \times \Omega \rightarrow S \quad \text{t.c. } X_t \text{ v.a. } \forall t \in \mathbb{R}. \quad (1.2)$$

Processi di questo genere sono atti a descrivere fenomeni aleatori che variano in modo continuo nel tempo, ad esempio il prezzo di mercato di un titolo durante una sessione di mercato.

Si distinguono i processi stocastici anche in base alla cardinalità dello spazio  $S$ , ovvero lo spazio dei valori assumibili dalle singole variabili aleatorie:

- $S$  continuo, ad esempio  $\mathbb{R}$ ;
- $S$  discreto, ad esempio  $\mathbb{N}$ .

Nella trattazione si prenderanno in considerazione prevalentemente i processi stocastici a tempo discreto e con spazio degli stati  $S$  finito, o al più numerabile.

*Osservazione 1.* Essendo  $S$  per ipotesi finito o numerabile, si possono enumerare gli stati e considerare  $X$  come processo stocastico a valore in "indice degli stati": ad esempio con  $(X_t = i)$  si può indicare che al tempo  $t$  il processo visita l'  $i$ -esimo stato.

*Osservazione 2.* Nell' ipotesi di  $S$  finito, per ogni istante  $n$  è possibile definire la distribuzione di probabilità di  $X_n$  tramite il vettore di componenti

$$\rho_i = \mathbb{P}(X_n = i) \quad \forall i \in S; \quad (1.3)$$

Una ulteriore ipotesi da assegnare al processo stocastico si basa sulla richiesta di indipendenza stocastica tra le variabili che la compongono; questa ipotesi tuttavia risulterebbe troppo restrittiva: ad esempio se  $X_n$  rappresenta il prezzo di un titolo finanziario alla chiusura dell'  $n$ -esimo giorno di mercato, difficilmente si potrebbe pensare a questa quantità come indipendente dal valore assunto il giorno precedente.

Una soluzione consiste nel considerare le variabili in generale dipendenti soltanto dalla variabile strettamente precedente, e indipendente dalle altre.

Un processo che rispetti questo principio è detto *Markoviano*.

**Definizione 1.1** (Catena di Markov). Dato uno spazio  $S$  finito o numerabile, un processo stocastico a tempo discreto e a valori in  $S$ ,  $(X_n)_{n \in \mathbb{N}}$ , è detto catena (o processo) di Markov omogenea se sono definiti:

- $(\pi_s)_{s \in S}$  distribuzione iniziale del processo:

$$\pi_i := \mathbb{P}(X_1 = i); \quad (1.4)$$

- $P = (p_{ij})_{i,j \in S}$  matrice di transizione del processo tale che

$$p_{i,j} \in [0, 1], \quad \sum_{j \in S} p_{i,j} = 1 \quad (1.5)$$

e se vale

$$\mathbb{P}(X_1 = x_1, X_2 = x_2, \dots, X_n = x_n) = \pi_{x_1} p_{x_1, x_2} \cdots p_{x_{n-1}, x_n}. \quad (1.6)$$

**Proposizione 1.2** (Proprietà di Markov). *Dato un processo di Markov come nella Definizione 1.1, si ha*

$$\mathbb{P}(X_n = x_n | X_0 = x_0, \dots, X_{n-1} = x_{n-1}) = \mathbb{P}(X_n = x_n | X_{n-1} = x_{n-1}) = p_{x_{n-1}, x_n} \quad (1.7)$$

Si osserva quindi che

- la variabile aleatoria al tempo  $n$  è indipendente dalle variabili "passate" eccetto che per quella immediatamente precedente (proprietà di Markov);
- la probabilità  $\mathbb{P}(X_n = j | X_1 = x_1, \dots, X_{n-1} = i) = p_{i,j}$  in particolare questa non dipende dal tempo considerato, ma solamente dagli stati (omogeneità).

#### 41. Catene di Markov omogenee con una infinità numerabile di stati.

*Esempio 1.* Come fenomeno aleatorio da esaminare si consideri la condizione meteo giornaliera di una determinata città nel corso di giugno. Per semplicità supponiamo che le possibili condizioni siano *sereno*, *piovoso*.

Si suppone che

- se la giornata  $n$  gode di un tempo sereno, con probabilità  $p$  la giornata successiva sarà serena, altrimenti (con probabilità  $1 - p$ ) pioverà;
- se il giorno  $n$  è piovoso, con probabilità  $q$  il giorno seguente si avrà bel tempo, mentre con probabilità  $1 - q$  continuerà a piovere.

Inoltre si fissa una distribuzione di probabilità iniziale sulle condizioni meteo del primo giorno in esame:

- con probabilità  $\rho_1$  il primo del mese è soleggiato;
- con probabilità  $\rho_2$  il primo giorno del mese è piovoso.

A questo punto il processo stocastico a tempo discreto  $(X)_n$ ,  $n = 1, 2, \dots, 30$ , dove  $X_i =$  condizione meteo nel giorno  $i$  giugno, è un processo di Markov di distribuzione iniziale  $\rho = (\rho_1, \rho_2)$  e matrice di transizione

$$P = \begin{pmatrix} p & 1 - p \\ q & 1 - q \end{pmatrix}$$

## Capitolo 2

# Modelli di Markov Nascosti

Consideriamo una catena di Markov  $(X_n)_{n \in \mathbb{N}}$  con spazio degli stati  $S$ , distribuzione iniziale  $\pi$  e matrice di transizione  $P$ .

Sia  $V = (v_1, \dots, v_M)$  un insieme finito di elementi, chiamati *segnali*: ogni qualvolta la catena  $X$  entra in uno stato viene emesso un segnale con probabilità dipendente esclusivamente dallo stato attuale della catena; più precisamente chiamato  $O_n$  il segnale emesso al tempo  $n$ , si ha che

$$\mathbb{P}(O_n = o_n | X_n = x_n, \dots, X_0 = x_0) = \mathbb{P}(O_n = o_n | X_n = x_n). \quad (2.1)$$

Inoltre si supporrà che la probabilità di emissione di un determinato elemento di  $V$  non dipenda dal tempo (ipotesi di stazionarietà), ma solo dallo stato in cui si trova la catena  $X$ :

$$\mathbb{P}(O_n = \bar{o} | X_n = \bar{x}) = \mathbb{P}(O_i = \bar{o} | X_i = \bar{x}) := b_{\bar{x}}(\bar{o}). \quad (2.2)$$

In ultima istanza si richiede che un segnale venga sempre emesso:

$$\sum_{k=1}^M b_x(v_k) := \sum_{k=1}^M b_x(k) = 1 \quad \forall x \in S. \quad (2.3)$$

*Osservazione 3.* La successione di segnali emessi è essa stessa un processo stocastico discreto a valori in uno spazio finito. Quindi anche in questo caso è possibile enumerare gli elementi di  $V$  e denotare l'evento  $(O_t = v_k) := (O_t = k)$ .

**Definizione 2.1.** Un modello come quello appena descritto, nel quale si osserva una sequenza di segnali  $(O_n)_n$  emessa da una catena di Markov  $X$  non direttamente osservabile, viene chiamato **Modello di Markov Nascosto**. Si può riassumere un modello di Markov nascosto come una coppia di processi stocastici  $(X_t, O_t)$ ,  $t \in \mathbb{N}$  e una terna di parametri  $\lambda = (\pi, P, B)$  dove

- $\pi$  è la distribuzione iniziale della catena di Markov  $X$ ;
- $P$  è la matrice di transizione del processo nascosto  $X$ ;
- $B = (b_i(k))$ ,  $i = 1, \dots, N$ ;  $k = 1, \dots, M$ , è la matrice delle probabilità di emissione, i cui elementi sono definiti in (2.2):

$$b_i(k) = b_i(v_k) = \mathbb{P}(O_t = v_k | X_t = i).$$

Si richiede inoltre che valgano le condizioni espresse dalle equazioni (2.1), (2.2), (2.3).

*Osservazione 4.* Il processo  $O$  delle osservazioni in generale non è una catena di Markov.

*Esempio 2.* Consideriamo un esempio calcistico: ogni rigore può essere battuto a destra del portiere (stato 1), a sinistra (stato 2) o centrale (stato 3). In ogni squadra che si rispetti c'è un giocatore specializzato nelle realizzazioni dal dischetto, chiamato rigorista. In questo modello consideriamo il rigorista della nostra squadra e supponiamo che esso scelga la traiettoria solo in base a dove ha tirato il rigore precedente.

Ad esempio se aveva tirato a destra precedentemente, tirerà di nuovo a destra con una probabilità dell' 80% , a sinistra con una probabilità del 15% e centrale al 5%. Ripetendo il discorso anche per le battute a sinistra e centrale si costruisce la matrice di transizione del processo stocastico  $X$  delle direzioni dei rigori calciati:

$$P = \begin{pmatrix} 0.8 & 0.15 & 0.5 \\ 0.7 & 0.2 & 0.1 \\ 0.4 & 0.4 & 0.2 \end{pmatrix}.$$

Detto ciò, come noto, l'esito di un rigore può essere *realizzato* (r) o *sbagliato* (s). Supponiamo di sapere che se il rigore è tirato a destra entrerà in porta con una probabilità del 90%, se è indirizzato a sinistra con probabilità 85% e se centrale con una probabilità dell' 80%: nel formalismo usato precedentemente

$$\begin{aligned} b_0(r) &= 0.9 & b_1(s) &= 0.1 \\ b_1(r) &= 0.85 & b_1(s) &= 0.15 \\ b_3(r) &= 0.8 & b_3(s) &= 0.2 \end{aligned}$$

A questo punto si può definire il modello di Markov nascosto, nel quale veniamo a conoscenza degli esiti dei rigori, ad esempio leggendo il resoconto delle partite sui giornali, senza sapere però dove sono stati calciati.

**Notazione 1.** In seguito si userà la seguente notazione per indicare i vettori, come in [1]:

$$\mathbf{x}_n = (x_1, \dots, x_n), \quad x_i \in S; \quad (2.4)$$

$$\mathbf{o}_n = (o_1, \dots, o_n), \quad o_i \in V \quad (2.5)$$

Allo stesso modo con  $\mathbf{X}_n = (X_1, \dots, X_n)$  e  $\mathbf{O}_n = (O_1, \dots, O_n)$  si indicheranno i vettori stocastici.

Dato un modello di Markov nascosto, si può trovare risposta a tre problemi fondamentali:

- *problema della valutazione*, o *evaluation problem*: calcolare la probabilità di osservazione di una fissata sequenza  $\mathbf{O}_n = (o_1, o_2, \dots, o_n)$ .
- *problema della decodifica*: fissate le osservazioni  $\mathbf{O}_n = (o_1, o_2, \dots, o_n)$ , stabilire quale è la sequenza più probabile di stati nascosti  $(X_1, \dots, X_n)$ .
- *problema dell' addestramento* o *learning problem*: data una sequenza di osservazioni, trovare un nuovo modello che si adatti meglio ai dati, ovvero modificare i valori di  $(\pi, P, B)$  in modo da rendere massima la probabilità di uscita dei segnali osservati.

## 2.1 Problema della valutazione.

Sia  $\lambda = (\pi, P, B)$  un modello di Markov nascosto e  $\mathbf{O}_n = (O_1, \dots, O_n)$  il vettore aleatorio dei primi  $n$  segnali emessi.

Fissata una successione  $\mathbf{o}_n = (o_1, \dots, o_n)$ ,  $o_1, \dots, o_n \in V$ , si cerca di stabilire

$$\mathbb{P}(\mathbf{O}_n = \mathbf{o}_n). \quad (2.6)$$

Ovviamente tale probabilità può essere calcolata considerando tutte le sequenze di  $n$  stati possibili:

$$\begin{aligned} \mathbb{P}(\mathbf{O}_n = \mathbf{o}_n) &= \sum_{i_1, \dots, i_n} \mathbb{P}(\mathbf{O}_n = \mathbf{o}_n | X_1 = i_1, \dots, X_n = i_n) \times \\ &\quad \times \mathbb{P}(X_1 = i_1, \dots, X_n = i_n) = \\ &= \sum_{i_1, \dots, i_n} b_{i_1}(o_1) \cdots b_{i_n}(o_n) p_{i_1, i_2} \cdots p_{i_{n-1}, i_n}. \end{aligned} \quad (2.7)$$

Tuttavia la formula (2.7) ha un'alto costo computazionale: la sommatoria è costituita da  $N^n$  elementi, ognuno dei quali è calcolato come prodotto di  $2n$  fattori. Il costo complessivo dunque è di  $2nN^n$ .

Il costo computazionale per la risoluzione del problema della valutazione può essere abbassato adottando due algoritmi, basati entrambi sulla somma di  $nN$  quantità, ognuna delle quali composta da  $2N + 1$  operazioni, ottenendo quindi algoritmi con costo computazionale  $\mathcal{O}(2nN^2)$  al posto di  $\mathcal{O}(2nN^n)$ .

### 2.1.1 Algoritmo in avanti.

Dato il vettore  $\mathbf{o}_n = (o_1, \dots, o_n)$ , si definisce

$$\alpha_n(j) := \mathbb{P}(\mathbf{O}_n = \mathbf{o}_n, X_n = j) \quad (2.8)$$

ovvero la probabilità che si osservino i segnali  $\mathbf{o}_n$  nei primi  $n$  tempi e che  $X$  sia nello stato  $j$  al tempo  $n$ . Osserviamo che

$$\mathbb{P}(\mathbf{O}_n = \mathbf{o}_n) = \sum_{j \in S} \mathbb{P}(\mathbf{O}_n = \mathbf{o}_n, X_n = j) = \sum_{j \in S} \alpha_n(j) \quad (2.9)$$



Inoltre si ha che

$$\mathbb{P}(X_n = j | \mathbf{O}_n = \mathbf{o}_n) = \frac{\alpha_n(j)}{\mathbb{P}(\mathbf{O}_n = \mathbf{o}_n)} = \frac{\alpha_n(j)}{\sum_{i \in S} \alpha_n(i)} \quad (2.10)$$

Dalla equazione (2.9) si evince che il problema della valutazione è risolto semplicemente calcolando le quantità  $\alpha_n(j)$ ,  $\forall j \in S$ . Si procederà per induzione, osservando il legame delle  $\alpha_n$  con le quantità  $\alpha_{n-1}$ :

$$\begin{aligned} \alpha_n(j) &= \mathbb{P}(\mathbf{O}_{n-1} = \mathbf{o}_{n-1}, O_n = o_n, X_n = j) = & (2.11) \\ &= \sum_{i \in S} \mathbb{P}(\mathbf{O}_{n-1} = \mathbf{o}_{n-1}, O_n = o_n, X_{n-1} = i, X_n = j) = \\ &= \sum_{i \in S} \mathbb{P}(\mathbf{O}_{n-1} = \mathbf{o}_{n-1}, X_{n-1} = i) \mathbb{P}(X_n = j, O_n = o_n | X_{n-1} = i, \mathbf{O}_{n-1} = \mathbf{o}_{n-1}) = \\ &= \sum_{i \in S} \alpha_{n-1}(i) \mathbb{P}(X_n = j, O_n = o_n | X_{n-1} = i) = \sum_{i \in S} \alpha_{n-1}(i) p_{i,j} b_j(o_n) = \\ &= b_j(o_n) \sum_{i \in S} \alpha_{n-1}(i) p_{i,j} \end{aligned} \quad (2.12)$$

Quindi in particolare

$$\alpha_n(j) = b_j(o_n) \sum_{i \in S} \alpha_{n-1}(i) p_{i,j}. \quad (2.13)$$

Si ottiene così l' algoritmo induttivo per il calcolo di  $\alpha_n(i), \forall i \in S$ :

- **inizializzazione:**

$$\alpha_1(i) = \mathbb{P}(X_1 = i, O_1 = o_1) = \pi_i b_i(o_1), \quad \forall i \in S \quad (2.14)$$

- **ricorsione:** si utilizza la formula (2.13); ad esempio per il calcolo degli  $\alpha_2(j), \forall j \in S$

$$\alpha_2(j) = b_j(o_2) \sum_{i \in S} \alpha_1(i) p_{i,j} \quad (2.15)$$

- **conclusione:** una volta calcolate le quantità  $\alpha_n$  si utilizza la formula (2.9) per risolvere il problema della valutazione.

### 2.1.2 Algoritmo in indietro

L'approccio in indietro (backward approach) consiste in un metodo alternativo per la risoluzione del problema della valutazione. Consiste nel definire le quantità  $\beta_k(i)$ , dati il modello di Markov  $\lambda$  e un vettore di osservazioni  $\mathbf{o}_n = (o_1, \dots, o_n)$ , come

$$\beta_k(i) := \mathbb{P}(O_{k+1} = o_{k+1}, \dots, O_n = o_n \mid X_k = i), \quad (2.16)$$

ovvero la probabilità di osservare i dati scelti dal tempo  $k + 1$  in poi, se la catena nascosta  $X$  visita lo stato  $i$  al tempo  $k$ . Come per l'approccio in avanti, si ottiene un algoritmo iterativo: condizionando  $\beta_k(i)$  alla variabile  $X_{k+1}$  si ottiene

$$\begin{aligned} \beta_k(i) &= \sum_{j \in S} \mathbb{P}(O_{k+1} = o_{k+1}, \dots, O_n = o_n \mid X_{k+1} = j, X_k = i) \times \\ &\quad \times \mathbb{P}(X_{k+1} = j \mid X_k = i) = \\ &= \sum_{j \in S} \mathbb{P}(O_{k+1} = o_{k+1}, \dots, O_n = o_n \mid X_{k+1} = j) p_{i,j} = \\ &= \sum_{j \in S} \mathbb{P}(O_{k+1} = o_{k+1} \mid X_{k+1} = j, X_k = i) \times \\ &\quad \times \mathbb{P}(O_{k+2} = o_{k+2}, \dots, O_n = o_n \mid X_{k+1} = j, O_{k+1} = o_{k+1}) p_{i,j} = \\ &= \sum_{j \in S} b_j(o_{k+1}) \beta_{k+1}(j) p_{i,j} \end{aligned} \quad (2.17)$$

Si osserva quindi la dipendenza di  $\beta_k(i)$  dai valori di  $\beta_{k+1}(j), \forall j \in S$ . Si implementa allora l'algoritmo:

- inizializzazione:

$$\beta_n(i) = 1 \quad \forall i \in S; \quad (2.18)$$

- iterazione: si calcola

$$\beta_k(i) = \sum_{j \in S} b_j(o_{k+1}) \beta_{k+1}(j) p_{i,j} \quad \forall i \in S \quad (2.19)$$

fino ad ottenere i valori di  $\beta_1(i)$ ,  $\forall i \in S$ .

A questo punto si risolve il problema osservando che

$$\begin{aligned} \mathbb{P}(\mathbf{O}_n = \mathbf{o}_n) &= \sum_{i \in S} \mathbb{P}(\mathbf{O}_n = \mathbf{o}_n | X_1 = i) \pi_i = \\ &= \sum_{i \in S} \mathbb{P}(O_1 = o_1 | X_1 = i) \mathbb{P}(O_2 = o_2, \dots, O_n = o_n | X_1 = i) \pi_i = \\ &= \sum_{i \in S} b_i(o_1) \beta_1(i) \pi_i \end{aligned} \quad (2.20)$$

### 2.1.3 Considerazioni computazionali.

In primo luogo osserviamo che gli approcci forward e backward sono equivalenti per la soluzione del problema. Tuttavia nel caso di osservazioni e spazi  $S$  e  $V$  composti di molti elementi, è preferibile utilizzare un metodo misto: osserviamo che fissato un tempo qualsiasi  $k \in \{1, \dots, n\}$  si ha

$$\begin{aligned} \mathbb{P}(\mathbf{O}_n = \mathbf{o}_n, X_k = j) &= \mathbb{P}(\mathbf{O}_k = \mathbf{o}_k, X_k = j) \times \\ &\quad \times \mathbb{P}(O_{k+1} = o_{k+1}, \dots, O_n = o_n | X_k = j) = \\ &= \alpha_k(j) \beta_k(j). \end{aligned} \quad (2.21)$$

ovvero, se si dispone delle quantità  $\alpha$  e  $\beta$  per uno stesso tempo, allora si può risolvere il problema calcolando

$$\begin{aligned} \mathbb{P}(\mathbf{O}_n = \mathbf{o}_n) &= \sum_{j \in S} \mathbb{P}(\mathbf{O}_n = \mathbf{o}_n, X_k = j) = \\ &= \sum_{j \in S} \alpha_k(j) \beta_k(j) \end{aligned} \quad (2.22)$$

quindi la strategia migliore sarebbe quella di effettuare un calcolo parallelo di  $\alpha_k$ , con  $k$  a partire da 1, e  $\beta_i$ , con  $i$  a partire da  $n$ , stoppare il calcolo quando si sono calcolate tutte le quantità per  $k = i$  e usufruire della relazione (2.22).

## 2.2 Problema della decodifica.

Il problema della decodifica (decoding problem) è senza dubbio uno dei problemi cardine dello studio dei modelli di Markov nascosti: fissata una sequen-

za di osservazioni  $\mathbf{o}_n = (o_1, \dots, o_n)$  e fissato un modello di Markov nascosto  $\lambda = (\pi, P, B)$ , si vuole determinare la sequenza di stati ottimale alla base delle osservazioni.

Il concetto di sequenza ottimale non è univocamente determinato, un primo criterio si basa sulla scelta di stati individualmente più probabili, ovvero per ogni  $t = 1, \dots, n$  si individua uno stato  $i_t$  tale che la probabilità

$$\gamma_t(i) = \mathbb{P}(X_t = i | \mathbf{O}_n = \mathbf{o}_n) \quad (2.23)$$

sia massima. Si osserva che

$$\gamma_t(i) = \frac{\mathbb{P}(X_t = i, \mathbf{O}_n = \mathbf{o}_n)}{\mathbb{P}(\mathbf{O}_n = \mathbf{o}_n)} = \frac{\alpha_t(i) \beta_t(i)}{\sum_{j \in S} \alpha_t(j) \beta_t(j)}. \quad (2.24)$$

Quindi lo stato cercato  $i_t$  è quello che massimizza  $\alpha_t(i) \beta_t(i)$ :

$$i_t = \arg \max_{i \in S} [\alpha_t(i) \beta_t(i)] \quad \forall t = 1, 2, \dots, n \quad (2.25)$$

*Osservazione 5.* Il risultato del procedimento (2.25) è la sequenza di stati  $(i_1, \dots, i_n)$  che massimizza il valore atteso di numero di stati esatti. Tuttavia il procedimento non tiene conto dei legami che intercorrono tra gli stati di una catena: ad esempio possono generarsi sequenze proibite dal modello (nel caso in cui due stati non comunicano).

### 2.2.1 Algoritmo di Viterbi.

Un secondo criterio di ottimalità per il problema della decodifica si basa sul principio di *best path*, ovvero si considera l'intera sequenza di stati  $(i_1, i_2, \dots, i_n)$ , tra le sequenze possibili, che rende massima

$$\mathbb{P}(X_1 = i_1, \dots, X_n = i_n | \mathbf{O}^n = \mathbf{o}_n), \quad (2.26)$$

e quindi equivale a trovare la sequenza che massimizza

$$\mathbb{P}(\mathbf{X}_n = (i_1, \dots, i_n), \mathbf{O}^n = \mathbf{o}_n). \quad (2.27)$$

Una soluzione iterativa per la ricerca del best path è stata ideata da Andrew Viterbi nel 1967, pubblicata in *Error bounds for convolutional codes*

and asymptotically optimum decoding algorithm, e illustrata di seguito.

Per  $t = 1, \dots, n$ , si definiscono le quantità

$$\delta_t(j) := \max_{i_1, \dots, i_{t-1}} \mathbb{P}(\mathbf{X}_{t-1} = (i_1, \dots, i_{t-1}), X_t = j, \mathbf{O}^t = \mathbf{s}_t), \quad (2.28)$$

ovvero la probabilità massima, al variare dei primi  $t - 1$  stati, di avere le prime  $t$  osservazioni date e la catena nascosta  $X$  nello stato  $j$  al tempo  $t$ .

Per come definito, vale

$$\begin{aligned} \delta_t(j) &= \max_i \max_{(i_1, \dots, i_{t-2})} \mathbb{P}(\mathbf{X}_{t-2} = (i_1, \dots, i_{t-2}), X_{t-1} = i, \\ &\quad X_t = j, O_t = o_t, \mathbf{O}^{t-1} = \mathbf{o}_{t-1}) = \\ &= \max_i \max_{(i_1, \dots, i_{t-2})} \mathbb{P}(\mathbf{X}_{t-2} = (i_1, \dots, i_{t-2}), X_{t-1} = i, \mathbf{O}^{t-1} = \mathbf{o}_{t-1}) p_{i,j} b_j(o_t) = \\ &= b_j(o_t) \max_i \delta_{t-1}(i) p_{i,j} \end{aligned} \quad (2.29)$$

Si osserva quindi l'opportunità di calcolo di queste probabilità tramite un algoritmo in avanti: per  $t = 1$  si definisce

$$\delta_1(j) = \mathbb{P}(X_1 = j, O_1 = o_1) = \pi_j b_j(o_1) \quad \forall j \in S; \quad (2.30)$$

A questo punto il valore massimo di  $\delta_1$  verrà utilizzato nel calcolo di  $\delta_2(j)$ ,  $\forall j \in S$ . Iterando il procedimento si calcolano le quantità  $\delta_t(i) \quad \forall t = \{1, \dots, n\}$ ,  $\forall i \in S$ .

In particolare si ha

$$\max_{i_1, \dots, i_n} \mathbb{P}(\mathbf{X}_n = (i_1, \dots, i_n), \mathbf{O}^n = \mathbf{o}_n) = \max_j \delta_n(j) \quad (2.31)$$

quindi, definito lo stato

$$j_n^* = \arg \max_{i \in S} [\delta_n(i)] \quad (2.32)$$

si ha che  $j_n^*$  è l'ultimo elemento della catena cercata.

A questo punto si procede a ritroso: si otterrà  $j_{n-1}^*$  da

$$\begin{aligned} &\max_{i_1, \dots, i_n} \mathbb{P}(\mathbf{X}_n = (i_1, \dots, i_n), \mathbf{O}^n = \mathbf{o}_n) = \delta_n(j_n^*) = \\ &= \max_{i_1, \dots, i_{n-1}} \mathbb{P}(\mathbf{X}_{n-1} = (i_1, \dots, i_{n-1}), X_n = j_n^*, \mathbf{O}^{n-1} = \mathbf{o}_{n-1}, O_n = o_n) = \\ &= \max_i \delta_{n-1}(i) p_{i, j_n^*} b_{j_n^*}(o_n). \end{aligned} \quad (2.33)$$

ovvero il penultimo stato  $j_{n-1}^*$  sarà

$$j_{n-1}^* = \arg \max_i [ p_{i,j_n^*} \delta_{n-1}(i) ]. \quad (2.34)$$

Iterando il procedimento appena illustrato si troveranno i restanti stati della catena ottimale. Questo algoritmo prende il nome di *Algoritmo di Viterbi*, e può essere riassunto nei seguenti passaggi:

- *Inizializzazione:*

$$\delta_1(i) = \pi_i b_i(o_1) \quad \forall i \in S \quad (2.35)$$

$$\psi_1(i) = 0 \quad \forall i \in S \quad (2.36)$$

- *Ricorsione in avanti:*

$$\delta_t(j) = \max_{i \in S} [\delta_{t-1}(i) p_{i,j}] b_j(o_t) \quad \forall t = 2, \dots, n \quad \forall i \in S \quad (2.37)$$

$$\psi_t(j) = \arg \max_{1 \leq i \leq N} [\delta_{t-1}(i) p_{i,j}] \quad (2.38)$$

- *Terminazione:*

$$P^* = \max_{j \in S} [\delta_n(j)] \quad (2.39)$$

$$j_n^* = \arg \max_i [\delta_n(i)] \quad (2.40)$$

- *Ricorsione all' indietro*

$$j_t^* = \psi_{t+1}(j_{t+1}^*), \quad t = n-1, \dots, 1. \quad (2.41)$$

- *soluzione:* il best path ottenuto è  $(j_1^*, \dots, j_n^*)$ .

## 2.3 Problema dell' apprendimento

In molte applicazioni, ad esempio nel riconoscimento vocale di parole, si richiede che il modello di Markov nascosto si adatti alle osservazioni. Questo problema è noto con il nome di problema dell' apprendimento, o learning problem.

Lo scopo di questa sezione è quello di trovare, dato un modello di Markov nascosto  $\lambda = (\pi, P, B)$  e un vettore di osservazioni  $\mathbf{o}_T = (o_1, \dots, o_T)$ , un nuovo modello  $\bar{\lambda} = (\bar{\pi}, \bar{P}, \bar{B})$  tale da massimizzare la probabilità della sequenza osservata.

Per distinguere il modello secondo il quale verrà calcolata la probabilità, si userà la notazione seguente:

**Notazione 2.** Si indicherà con

$$\mathbb{P}(\cdot | \lambda) \quad (2.42)$$

la misura di probabilità dato il modello  $\lambda$ .

In definitiva si cercherà  $\bar{\lambda}$  in modo che

$$\mathbb{P}(\mathbf{O}_T = \mathbf{o}_T | \bar{\lambda}) = \max_{\lambda^*} \mathbb{P}(\mathbf{O}_T = \mathbf{o}_T | \lambda^*). \quad (2.43)$$

Una soluzione al problema è data dall' algoritmo di Baum-Welch, che consiste nel ristimare i parametri con

$$\bar{p}_{ij} = \frac{\sum_{t=1}^{T-1} \alpha_t(i) p_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\sum_{t=1}^{T-1} \alpha_t(i) \beta_t(i)} \quad (2.44)$$

$$\bar{b}_j(k) = \frac{\sum_{t \ni \{o_t=k\}} \alpha_t(j) \beta_t(j)}{\sum_{t=1}^T \alpha_t(j) \beta_t(j)} \quad (2.45)$$

$$\bar{\pi}_i = \frac{\alpha_1(i) \beta_1(i)}{\mathbb{P}(\mathbf{o}_T | \lambda)} \quad (2.46)$$

con  $\mathbb{P}(\mathbf{o}_T | \lambda)$  calcolabile con un algoritmo in avanti (2.17), o all' indietro (2.20).

Si può dare alle formule di ristima una accezione statistica: infatti definiamo

$$\xi_t(i, j) = \mathbb{P}(X_t = i, X_{t+1} = j | \mathbf{O}_T = \mathbf{o}_T, \lambda) \quad (2.47)$$

ovvero la probabilità di passare dallo stato  $i$  allo stato  $j$ , fissato il modello originario  $\lambda$  e le osservazioni  $\mathbf{o}_T$ . Per le proprietà della attesa condizionata si ha che

$$\begin{aligned}\xi_t(i, j) &= \frac{\mathbb{P}(X_t = i, X_{t+1} = j, \mathbf{O}_T = \mathbf{o}_T \mid \lambda)}{\mathbb{P}(\mathbf{O}_T = \mathbf{o}_T \mid \lambda)} = \\ &= \frac{\alpha_t(i)P_{ij}b_j(o_{t+1})\beta_{t+1}(j)}{\mathbb{P}(\mathbf{O}_T = \mathbf{o}_T \mid \lambda)}\end{aligned}$$

Sommando sui tempi  $t = 1, \dots, T - 1$  si otterrà quindi il valore atteso del numero di transizioni dallo stato  $i$  allo stato  $j$ .

Allo stesso modo si calcola la probabilità di essere nello stato  $i$  al tempo  $t$  dato il vettore di osservazioni  $\mathbf{o}_T$

$$\gamma_t(i) = \mathbb{P}(X_t = i \mid \mathbf{O}_T = \mathbf{o}_T, \lambda) = \sum_{j=1}^T \xi_t(i, j) \quad (2.48)$$

e di conseguenza il numero atteso di visite allo stato  $i$

$$\gamma(i) = \sum_{t=1}^{T-1} \gamma_t(i). \quad (2.49)$$

A questo punto le ristime dei parametri date da (2.44),(2.45),(2.46), possono essere viste come:

$$\bar{\pi}_i = \text{numero atteso di visite } i\text{-esimo stato al tempo}(t = 1) = \gamma_1(i)$$

$$\bar{p}_{ij} = \frac{\text{numero atteso transizioni da } i \text{ a } j}{\text{numero atteso di visite in } i} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

$$\bar{b}_i(o_k) = \frac{\text{numero atteso di visite in } j \text{ con emissione di } o_k}{\text{numero atteso di visite in } j}$$



### 2.3.1 Prova delle formule di ristima.

Siano dati un modello di Markov nascosto  $\lambda = (\pi, P, B)$  e un vettore di osservazioni di lunghezza  $T$ ,  $\mathbf{o}_T = (o_1, \dots, o_T)$ , a componenti nello spazio finito dimensionale  $V = (v_1, \dots, v_M) := (1, \dots, M)$ .

Nella dimostrazione della validità delle formule di ristima, sarà utile considerare lo spazio delle sequenze di stati di lunghezza  $T$ , i cui elementi saranno quindi dei vettori  $\mathbf{x}_T = (x_1, \dots, x_T)$ .

Essendo lo spazio degli stati finito, l' insieme delle  $T$ - sequenze appena menzionato sarà finito, e sarà utile ai fini della dimostrazione numerarle:

**Notazione 3.** Consideriamo lo spazio delle sequenze di  $T$  stati: denotiamo con

$$u_i = \mathbb{P}(\text{i-esima sequenza, osservazioni } \mathbf{O}_T = \mathbf{o}_T | \lambda) = \mathbb{P}(i, \mathbf{o}_T | \lambda) \quad (2.50)$$

la probabilità congiunta della  $i$ -esima sequenza e delle osservazioni date nel modello di Markov nascosto  $\lambda$ . Allo stesso modo

$$v_i = \mathbb{P}(\text{i-esima sequenza, osservazioni } \mathbf{O}_T = \mathbf{o}_T | \bar{\lambda}) = \mathbb{P}(i, \mathbf{o}_T | \bar{\lambda}). \quad (2.51)$$

*Osservazione 6.* Date le quantità  $u_i$  e  $v_i$ , si ha che

$$\begin{aligned} \mathbb{P}(\mathbf{o}_T | \lambda) &= \sum_i u_i \\ \mathbb{P}(\mathbf{o}_T | \bar{\lambda}) &= \sum_i v_i. \end{aligned}$$

Si useranno le quantità appena introdotte applicandole al teorema seguente :

**Teorema 2.2.** *Siano  $u_i$ ,  $i = \{1, \dots, S\}$ , numeri reali positivi, e  $v_i$ ,  $i = \{1, \dots, S\}$ , un insieme di numeri reali non negativi tali che*

$$\sum_{i=1}^S v_i > 0.$$

Allora si ha che

$$\log\left(\frac{\sum v_i}{\sum u_i}\right) \geq \frac{1}{\sum_k u_k} \left[ \sum_i (u_i \log v_i - u_i \log u_i) \right] \quad (2.52)$$

con sommatorie tutte da 1 a  $S$ .

*Dimostrazione.* Il teorema è diretta conseguenza della concavità della funzione logaritmo:

$$\begin{aligned} \log\left(\frac{\sum v_i}{\sum u_i}\right) &= \log\left[\sum_i \left(\frac{u_i}{\sum_k u_k}\right) \frac{v_i}{u_i}\right] \geq \sum_i \frac{u_i}{\sum_k u_k} \log\left(\frac{v_i}{u_i}\right) = \\ &= \frac{1}{\sum_k u_k} \left[ \sum_i (u_i \log v_i - u_i \log u_i) \right] \end{aligned} \quad (2.53)$$

□

Si osserva facilmente che il Teorema 2.2, utilizzando le quantità  $u_i$  e  $v_i$  introdotte nella Notazione 3, implica che

$$\log\left(\frac{\mathbb{P}(\mathbf{o}_T|\bar{\lambda})}{\mathbb{P}(\mathbf{o}_T|\lambda)}\right) \geq \frac{1}{\mathbb{P}(\mathbf{o}_T|\lambda)} [Q(\lambda, \bar{\lambda}) - Q(\lambda, \lambda)] \quad (2.54)$$

dove  $Q(\lambda, \bar{\lambda}) = \sum_i u_i \log v_i$ .

Quindi, se  $\bar{\lambda}$  è un modello tale che  $Q(\lambda, \bar{\lambda}) - Q(\lambda, \lambda) > 0$ , allora si ha

$$\frac{\mathbb{P}(\mathbf{o}_T|\bar{\lambda})}{\mathbb{P}(\mathbf{o}_T|\lambda)} > 1 \Rightarrow \mathbb{P}(\mathbf{o}_T|\bar{\lambda}) > \mathbb{P}(\mathbf{o}_T|\lambda) \quad (2.55)$$

ovvero la sequenza  $\mathbf{o}_T$  è più probabile nel modello di Markov nascosto  $\bar{\lambda}$ , il che corrisponde alla soluzione del problema dell' apprendimento.

Ovviamente la soluzione ottimale si ottiene trovando

$$\bar{\lambda} = \arg \max [Q(\lambda, \bar{\lambda}) - Q(\lambda, \lambda)]; \quad (2.56)$$

ciò si otterrà applicando la tecnica dei *moltiplicatori di Lagrange*. In particolare si dimostra che

**Teorema 2.3.** *Date delle quantità  $c_i > 0$ ,  $i = 1, \dots, N$ , la funzione*

$$F(\mathbf{x}) = F(x_1, \dots, x_N) = \sum_{i=1}^N c_i \log x_i \quad (2.57)$$

*vincolata dalla condizione*

$$\sum_{i=1}^N x_i = 1 \quad (2.58)$$

*assume il valore massimo per*

$$x_i = \frac{c_i}{\sum_i c_i}. \quad (2.59)$$

*Dimostrazione.* La prova consiste nell' applicare il metodo dei moltiplicatori di Lagrange alla funzione  $F$ : si considera la funzione lagrangiana

$$F(\mathbf{x}) - \mu \sum_i x_i = c_i \log x_i - \mu \sum_i x_i. \quad (2.60)$$

Risolvendo il sistema

$$\frac{\partial}{\partial x_i} \left[ F(\mathbf{x}) - \mu \sum_i x_i \right] = \frac{c_i}{x_i} - \mu = 0, \quad \forall i = 1, \dots, N \quad (2.61)$$

si ottiene

$$\mu = \sum_i c_i$$

semplicemente moltiplicando (2.61) per  $x_i$ , sommando su  $i$  e applicando la proprietà (2.58).

Si prova che il punto è un massimo per il fatto che:

$$\frac{\partial^2}{\partial x_i^2} F(\mathbf{x}) = -c_i \frac{1}{x_i^2} < 0 \quad \forall i = 1, \dots, N$$

□

Utilizzando il Teorema si avrà una prova diretta della bontà delle formule di ristima: consideriamo una generica sequenza di  $T$  stati, ad esempio la sequenza  $s$ -esima,  $\mathbf{s} = (s_1, \dots, s_T)$ . Dato il vettore di osservazioni  $\mathbf{o}_T$  si ha

che la probabilità congiunta di  $s$  e  $\mathbf{o}_T$  in un modello di Markov nascosto  $\bar{\lambda}$  è data da

$$v_s = \mathbb{P}(s, \mathbf{o}_T | \bar{\lambda}) = \bar{\pi}_{s_1} \bar{b}_{s_1}(o_1) \bar{p}_{s_1, s_2} \bar{b}_{s_2}(o_2) \cdots \bar{p}_{s_{T-1}, s_T} \bar{b}_{s_T}(o_T) \quad (2.62)$$

quindi

$$\log v_s = \log \bar{\pi}_{s_1} + \sum_{t=1}^{T-1} \log \bar{p}_{s_t, s_{t+1}} + \sum_{t=1}^T \log \bar{b}_{s_t}(o_t). \quad (2.63)$$

Sostituendo nella definizione di  $Q(\lambda, \bar{\lambda})$  e raccogliendo i termini uguali si ottiene

$$\begin{aligned} Q(\lambda, \bar{\lambda}) = \sum_{i=1}^N \sum_{j=1}^N c_{ij} \log \bar{p}_{ij} + \sum_{j=1}^N \sum_{k=1}^M d_{jk} \bar{b}_j(k) + \\ + \sum_{i=1}^N e_i \log \bar{\pi}_i \end{aligned} \quad (2.64)$$

in cui i coefficienti sono uguali a

$$c_{i,j} = \sum_{s=1}^S \mathbb{P}(s\text{-esima sequenza di } T \text{ stati, } \mathbf{o}_T | \lambda) n_{ij}(s)$$

$n_{ij}(s)$  = numero di transizioni dallo stato  $i$  allo stato  $j$  nella catena  $s$ -esima;

$$d_{jk} = \sum_{s=1}^S \mathbb{P}(s\text{-esima sequenza di } T \text{ stati, } \mathbf{o}_T | \lambda) m_{jk}(s)$$

$m_{jk}(s)$  = numero di volte nel quale viene emesso il  $k$ -esimo segnale dallo stato  $j$ ;

$$e_i = \sum_{s=1}^S \mathbb{P}(s\text{-esima sequenza di } T \text{ stati, } \mathbf{o}_T | \lambda) \delta_{s_1}^i$$

dove  $\delta_{s_1}^i$  è la delta di Kronecker, ovvero

$$\begin{aligned} \delta_{s_1}^i &= 1 \quad \text{se } s_1 = i; \\ &= 0 \quad \text{altrimenti.} \end{aligned}$$

A questo punto si applica il Teorema 2.3 alle espressioni (2.64), ottenendo il massimo di  $Q(\lambda, \bar{\lambda})$ :

$$\begin{aligned}\bar{p}_{ij} &= \frac{c_{ij}}{\sum_j c_{ij}} \\ \bar{b}_j(k) &= \frac{d_{jk}}{\sum_k d_{jk}} \\ \bar{\pi}_i &= \frac{e_i}{\sum_i e_i}\end{aligned}\tag{2.65}$$

e ciò equivale ad applicare le formule di ristima di Baum- Welch (2.44),(2.45), (2.46).

### 2.3.2 Formula di ristima con scaling.

Nelle formule di ristima di Baum-Welch sono coinvolte le variabili forward ( $\alpha$ ) e backward ( $\beta$ ). Tuttavia ci si accorge facilmente che entrambe le quantità tendono a valori prossimi allo zero molto velocemente. Ad esempio, ricordando la definizione della variabile forward (2.8):

$$\begin{aligned}\alpha_t(i) &= \mathbb{P}(\mathbf{o}_t, X_t = i | \lambda) = \sum_{X_1, \dots, X_{t-1}} \mathbb{P}(\mathbf{o}_t, X_t = i | X_1, \dots, X_{t-1}) \mathbb{P}(X_1, \dots, X_{t-1}) = \\ &= \sum_{X_1, \dots, X_{t-1}} \left[ \prod_{s=1}^t b_{X_t}(o_s) \prod_{s=1}^{t-1} p_{X_s, X_{s+1}} \right].\end{aligned}\tag{2.66}$$

essendo le produttorie della formule (2.66) composte da numeri minori di 1, per  $t$  grande abbastanza le variabili forward divengono tutte prossime a 0. Numericamente ciò porta all' inefficacia dell' algoritmo, in quanto si annullerebbero tutte le quantità dei parametri ( $P, b, \pi$ ).

Una soluzione è data dallo *scaling* di queste quantità. Il procedimento consiste nel moltiplicare di volta in volta per quantità dipendenti solo dal tempo, effettuare l' algoritmo di Baum- Welch, e successivamente annullare l'effetto dello scaling.

Una strategia di scaling consiste nello schema seguente:

- inizializzazione:

$$\begin{aligned}\alpha'_1(i) &= \alpha_1(i) \quad \forall i = 1, \dots, N; \\ c_1 &= \frac{1}{\sum_{i=1}^N \alpha'_1(i)}; \\ \bar{\alpha}_1(i) &= c_1 \alpha'_1(i);\end{aligned}$$

si osserva che  $\sum_i \bar{\alpha}_1(i) = 1$ .

- induzione

$$\begin{aligned}\alpha'_t(i) &= \sum_{j=1}^N \bar{\alpha}_{t-1}(j) p_{i,j} b_j(o_t) \quad \forall i = 1, \dots, N; \\ c_t &= \frac{1}{\sum_{i=1}^N \alpha'_t(i)}; \\ \bar{\alpha}_t(i) &= c_t \alpha'_t(i);\end{aligned}$$

Si può notare come le quantità di normalizzazione  $c_t$  dipendano solo dal tempo. Per induzione si ha che

$$\bar{\alpha}_T(i) = \left( \prod_{t=1}^T c_t \right) \alpha_T(i) \quad (2.67)$$

Usando le quantità scalate si ottiene che

$$\begin{aligned}1 &= \sum_{i=1}^N \bar{\alpha}_T(i) = \sum_{i=1}^N \left( \prod_{t=1}^T c_t \right) \alpha_T(i) = \\ &= \left( \prod_{t=1}^T c_t \right) \sum_{i=1}^N \alpha_T(i) = \left( \prod_{t=1}^T c_t \right) \mathbb{P}(\mathbf{o}_T | \lambda).\end{aligned} \quad (2.68)$$

Quindi  $\mathbb{P}(\mathbf{o}_T | \lambda) = \frac{1}{\prod_{t=1}^T c_t} := \frac{1}{\mathbf{c}_T}$ .

Per le variabili all' indietro si utilizza un processo di scaling simile:

- inizializzazione

$$\begin{aligned}\beta'_T(i) &= 1; \\ \bar{\beta}_T(i) &= c_T \beta'_T(i),\end{aligned}$$

- induzione

$$\begin{aligned}\beta'_t(i) &= \sum_{j=1}^N p_{i,j} b_j(o_{t+1}) \bar{\beta}_{t+1}(j); \\ \bar{\beta}_t(i) &= c_t \beta'_t(i),\end{aligned}$$

Dal procedimento si ottiene che

$$\bar{\beta}_t(i) = \left( \prod_{s=t}^T c_s \right) \beta_t(i) := \mathbf{D}_t \beta_t(i). \quad (2.69)$$

Si introducono a questo punto le quantità scalate nelle formule di ristima: ad esempio in (2.44)

$$\begin{aligned}\bar{p}_{ij} &= \frac{\sum_{t=1}^{T-1} \alpha_t(i) p_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\sum_{t=1}^{T-1} \alpha_t(i) \beta_t(i)} = \\ &= \frac{\sum_{t=1}^{T-1} \bar{\alpha}_t(i)/\mathbf{C}_t p_{ij} b_j(o_{t+1}) \bar{\beta}_{t+1}(j)/\mathbf{D}_{t+1}}{\sum_{t=1}^{T-1} \bar{\alpha}_t(i)/\mathbf{C}_t \bar{\beta}_t(i)/\mathbf{D}_t} = \\ &= \frac{(\sum_{t=1}^{T-1} \bar{\alpha}_t(i) p_{ij} b_j(o_{t+1}) \bar{\beta}_{t+1}(j))/\mathbf{C}_T}{(\sum_{t=1}^{T-1} \bar{\alpha}_t(i) \bar{\beta}_t(i)/c_t)/\mathbf{C}_T}\end{aligned}$$

Osservando che  $\mathbf{C}_t \cdot \mathbf{D}_{t+1} = \prod_{i=1}^N c_i = \mathbf{C}_T$ , e che  $\mathbf{C}_t \cdot \mathbf{D}_t = \prod_{i=1}^N c_i = \mathbf{C}_T \cdot c_t$ . In definitiva si ottiene la formula di ristima, nel quale ora sono in gioco valori  $\bar{\alpha}$  e  $\bar{\beta}$  che non decrescono velocemente a zero:

$$\bar{p}_{ij} = \frac{\sum_{t=1}^{T-1} \bar{\alpha}_t(i) p_{ij} b_j(o_{t+1}) \bar{\beta}_{t+1}(j)}{\sum_{t=1}^{T-1} \bar{\alpha}_t(i) \bar{\beta}_t(i)/c_t} \quad (2.70)$$

Allo stesso modo si ottiene che

$$\bar{b}_j(k) = \frac{\sum_{t \ni \{o_t=k\}} \bar{\alpha}_t(j) \bar{\beta}_t(j) / c_t}{\sum_{t=1}^T \bar{\alpha}_t(j) \bar{\beta}_t(j) / c_t}. \quad (2.71)$$

## 2.4 Algoritmi genetici applicati all' addestramento dei parametri.

Le formule di ristima di Baum- Welch presentate nella sezione precedente sono utili per addestrare i parametri di una catena di Markov nascosta, dato un vettore di osservazioni. Tuttavia il metodo presenta diversi inconvenienti:

- dipendenza sensibile dai valori iniziali dei parametri.
- scarsa efficienza per vettori di osservazioni non abbastanza lunghi.
- I modelli stimati con il metodo di Baum-Welch in genere tendono ad un modello di massimo locale, ovvero si tende al modello più vicino che massimizza la verosimiglianza delle osservazioni.

Spesso quindi è necessario, o quantomeno vantaggioso, utilizzare altri metodi per risolvere il problema dell' addestramento. Una possibilità è data dall' utilizzo degli *algoritmi genetici*.

Gli algoritmi genetici (AG) consistono in metodi di ricerca statistica basata sulle leggi naturali dell' evoluzione e della riproduzione. Furono introdotti da John Holland nel 1975 nello studio dell' evoluzione di sistemi adattivi naturali sotto l' ipotesi che competizione e innovazione fossero i principi cardine dell' adattamento (e quindi sopravvivenza) dei caratteri.

Nel corso degli anni gli AG sono stati applicati con successo in svariati campi, oltre che allo studio della biologia e della genetica, come ad esempio alla elettronica, all' economia e allo studio dell' intelligenza artificiale.

In [8] si può trovare una breve ma completa introduzione agli algoritmi genetici con svariate applicazioni pratiche. L' approccio usato per l' applicazione degli AG al learning problem per catene di Markov Nascoste è basato sui lavori di Kwong e Chau ([10]) e di Xiao ([9]).



### 2.4.1 Descrizione degli algoritmi genetici.

I soggetti sui quali agiscono gli algoritmi genetici sono sostanzialmente i  *cromosomi* , ovvero sequenze di valori a valori in uno spazio (ad esempio sequenze binarie, o sequenze di numeri reali). Un cromosoma quindi può essere descritto come un array di valori.

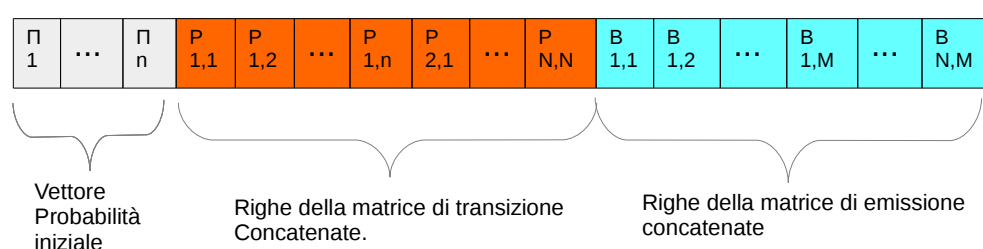


Figura 2.1: Esempio di cromosoma: in particolare si presenta la forma di cromosoma corrispondente ad un modello di Markov nascosto  $\lambda = (\Pi, P, B)$ .

L' algoritmo prende il via dalla generazione, nella maggior parte di casi randomica, di una popolazione di cromosomi di cardinalità fissata.

A questo punto la popolazione, tramite operazioni genetiche, si evolverà nel tempo formando di volta in volta una nuova popolazione che rimpiazza la precedente. Il ciclo riproduttivo termina quando nella popolazione è presente un cromosoma con le caratteristiche volute, o quando si raggiunge un limite sul numero di generazioni disposte a creare.

Più formalmente, per definire un algoritmo genetico  $c'$  è bisogno di:

- funzioni di *codifica* e *decodifica*, ovvero regole che associano agli elementi studiati dei cromosomi e viceversa. Ad esempio si possono codificare i numeri macchina in sequenze binarie, e decodificare le sequenze binarie in numeri macchina;

- *funzione di fitness*: una funzione sui cromosomi, o sulla loro decodifica, che associano ad ogni elemento un valore, legato alla "qualità" dell'elemento.

A questo punto, dopo aver generato la popolazione iniziale di cromosomi, si assegna ad ognuno di essi il valore di fitness corrispondente e può iniziare il processo evolutivo, che consiste in

1. *verifica delle condizioni di arresto*: in generale se un cromosoma o la sua decodifica hanno le proprietà cercate (indicate dal valore di fitness), o si è proceduto abbastanza a lungo;
2. *selezione*: tra la popolazione si scelgono in modo random degli elementi che parteciperanno alla creazione della generazione futura; un criterio particolarmente utile è quello della *regola della roulette*, che prevede l'utilizzo di una distribuzione di probabilità per la scelta di un elemento, proporzionale al valore di fitness del cromosoma (più il cromosoma è adatto e più ha probabilità di partecipare alla generazione futura);
3. *operazioni genetiche*: si estraggono, sempre con la regola della roulette, degli elementi che subiranno mutazioni:
  - *cross-over*: si estraggono due cromosomi e un numero random, se quest'ultimo è minore della *probabilità di cross-over* (fissata a priori), i due cromosomi creano altri due cromosomi scambiandosi delle informazioni. Più precisamente, dopo la scelta di uno o più punti di cross-over, si procede alla creazione di cromosomi figli mescolando i valori dei cromosomi genitori come in Figura 2.2.
  - *mutazione*: dopo l'operazione di cross-over, per ogni nuovo cromosoma si estrae a sorte un numero random; se quest'ultimo è minore della *probabilità di mutazione* (in genere un numero piccolo), al cromosoma verrà modificato in maniera casuale un elemento casuale.

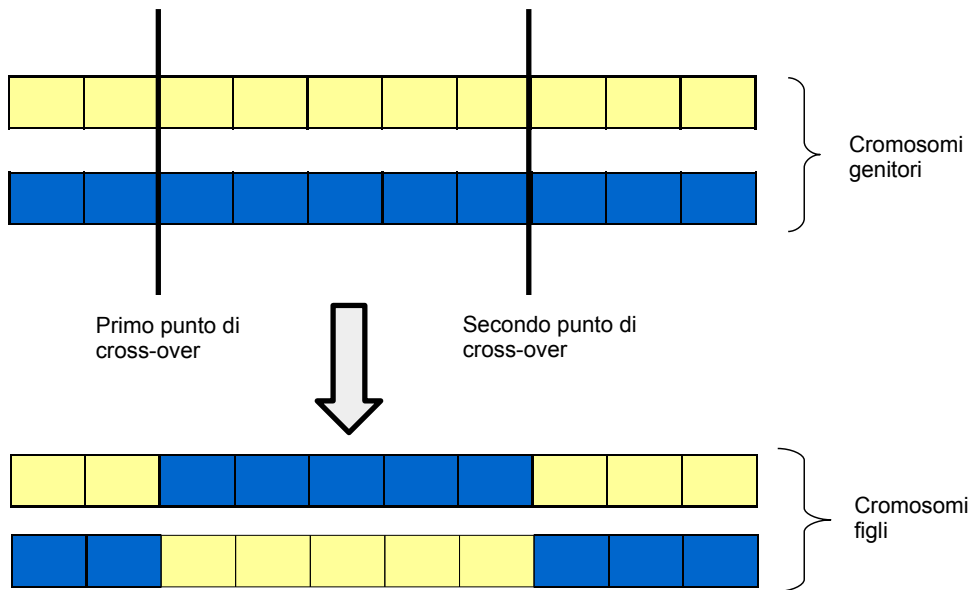


Figura 2.2: Esempio di cross-over a due punti.

4. *valutazione*: a questo punto si è creata una nuova generazione. Questi nuovi elementi devono essere valutati secondo la funzione di fitness e si procederà di nuovo dal punto 1.

Alla fine del ciclo ci sarà un candidato "migliore", ovvero il cromosoma con il valore di fitness più alto, che sarà adottato come soluzione o approssimazione del problema.

### 2.4.2 Applicazione al learning problem.

Applichiamo l' algoritmo genetico per l' addestramento dei parametri di una catena di Markov nascosta a partire dall' idea di Kwong pubblicata nel 1997 in *Optimization of HMM by genetic algorithm* ([10]). Come cromosomi si prenderanno in considerazione delle sequenze di numeri reali di lunghezza  $N + N^2 + N \cdot M$ , dove come fatto precedentemente,  $N$  è il numero degli stati della catena di Markov e  $M$  è la cardinalità dell' insieme dei possibili segnali. In particolare, dato un modello di Markov nascosto  $\lambda = (P, b, \pi)$ , si può

codificare il modello in un cromosoma mediante la concatenazione successiva di  $\pi$ , delle righe di  $P$  e delle righe di  $b$ , come in Figura 2.1.

Come funzione di fitness si può considerare semplicemente la probabilità del vettore di osservazioni di addestramento  $\mathbf{o}_T$  nel modello. Il valore di fitness di un cromosoma quindi può essere calcolato, dopo la decodifica, con un algoritmo in avanti:

$$f_n = \mathbb{P}(\mathbf{o}_T | \lambda_n) \quad (2.72)$$

dove  $\lambda_n = (P_n, b_n, \pi_n)$  sta ad indicare l'  $n$ -esimo elemento della popolazione. Quindi il problema dell' addestramento si conclude prendendo il cromosoma con valore di fitness più alto dell' ultima generazione, e successivamente decodificarlo.

E' bene specificare che, essendo i cromosomi effettivamente divisi in 3 parti ( il vettore  $\pi$ , le righe di  $P$  e le righe di  $b$ ), le operazioni di cross-over tra due cromosomi andranno fatte anche esse "a blocchi": si estraggono delle posizioni di cross-over per ogni blocco e i rimescolamenti saranno fatti solo tra parti corrispondenti.

*Osservazione 7.* E' molto importante specificare che nelle operazioni di *mutazioni* e *cross over*, si creano in generale dei cromosomi la cui decodifica non rappresenta un HMM.

Per ovviare il problema occorre effettuare le normalizzazioni dei parametri dopo ogni operazione genetica, questo procedimento può essere svolto semplicemente decodificando il cromosoma, riscaldando le quantità e infine ricodificare nella sequenza.

*Osservazione 8.* La soluzione del learning problem con algoritmi genetici non necessita di stime iniziali dei parametri  $(P, b, \pi)$ ; ciò può rappresentare un notevole vantaggio qualora si parta da una condizione di non informazione; inoltre l'algoritmo va alla ricerca di un massimo globale per la funzione di fitness, ovvero si cerca il modello in assoluto migliore che spiega le osservazioni.

# Capitolo 3

## Giochi di Markov

### 3.1 Cenni di teoria dei giochi

La teoria dei giochi è la scienza che studia le situazioni di interazione tra più agenti, o decisori, che si trovano a dover scegliere tra un ventaglio di azioni possibili al fine di raggiungere un determinato obiettivo. Il nome deriva dal libro di von Neumann e Morgenstern *Theory of Games and Economic Behaviour*, del 1944.

In questa sezione si richiameranno alcuni elementi della teoria dei giochi, in particolare gli aspetti che saranno utili nel proseguo della trattazione.

Un gioco per definizione è una situazione in cui due o più agenti si trovano a dover scegliere una azione, in modo simultaneo o consequenziale, e in base alle scelte fatte ognuno di loro avrà una ricompensa.

La prima distinzione nella classificazione dei giochi è da fare tra

- giochi non cooperativi, nei quali i giocatori non possono, o non vogliono, mettersi d' accordo per ottimizzare i loro guadagni;
- giochi cooperativi, nei quali sono possibili accordi vincolanti tra gli agenti.

Nel proseguo della trattazione si analizzeranno solo giochi *non cooperativi*, e a *somma zero*, ovvero giochi nei quali si può pensare che la vittoria di un

giocatore corrisponda esattamente alla perdita del suo avversario.

Di particolare utilità è la cosiddetta *forma normale* di un gioco, ovvero un modo sintetico di presentare il gioco nella sua interezza: siano

- l'insieme di  $N$  giocatori  $\{1, \dots, N\}$
- $\Sigma_1, \dots, \Sigma_N$  gli insiemi (non vuoti) contenenti le possibili azioni di ogni giocatore;
- $f_1, f_2, \dots, f_N$  funzioni *payoff* definite sul prodotto cartesiano degli insiemi  $\Sigma_i$ :

$$f_i : \prod_{k=1}^N \Sigma_k \rightarrow \mathbb{R} \quad \forall i = 1, \dots, N \quad (3.1)$$

dove  $f_i$  può essere interpretata come la funzione che associa il guadagno del giocatore  $i$  ad ogni possibile combinazione di mosse dei giocatori.

Se il gioco è tra 2 giocatori, come nei casi che si affronteranno da qui in avanti, un modo sintetico di descrivere il gioco è tramite la *matrice di payoff*, ovvero una matrice  $A$ , la cui entrata  $a_{i,j} = f_1(i, j)$ .

*Esempio 3.* Un esempio di gioco a due giocatori, non cooperativo e a somma zero in forma normale può essere descritto nel seguente modo: due giocatori, giocatore1 e giocatore2, sono uno di fronte all'altro; il giocatore1 può lanciare una palla a destra (D) o sinistra (S) e allo stesso modo il giocatore2 può lanciarsi a destra o sinistra per bloccarla.

Supponiamo che, se la scelta del tiratore e del ricevitore coincide, allora il tiratore paga una moneta al ricevitore, viceversa è il ricevitore a doverla al giocatore1.

Ovvero le funzioni di payoff sono

$$f_1(D, D) = f_1(S, S) = -1 \quad (\text{I perde una moneta}) \quad (3.2)$$

$$f_1(D, S) = f_1(S, D) = 1 \quad (\text{I vince una moneta}) \quad (3.3)$$

Ovviamente la funzione per il giocatore2 sarà dello stesso tipo ma di segni opposti. Quindi possiamo considerare solo una funzione di payoff e porre come obiettivo del giocatore1 quello di massimizzare il risultato della funzione payoff, mentre per il giocatore2 di minimizzarlo.

In questo caso quindi si può riassumere il gioco con la matrice di gioco

$I/II$	$D$	$S$
$D$	-1	1
$S$	1	-1

Ovvero a seconda delle mosse scelte dai giocatori 1 e 2 ( chiamati anche I e II) si seleziona il payoff corrispondente all' entrata della matrice.

*Osservazione 9.* Ovviamente nell' esempio3, i giocatori a priori non hanno individualmente una scelta più giusta delle altre, ovvero razionalmente non possono preferire una mossa rispetto all' altra. Diverso sarebbe il caso in cui la matrice dei payoff fosse

$I/II$	$D$	$S$
$D$	-1	0
$S$	1	-1

In questo caso il giocatore II non ha interesse alcuno nel preferire la mossa (D), in quanto con la mossa (S) non può perdere mai. Quindi l' unica scelta sensata del giocatore II è quella di buttarsi sempre a sinistra.

Allo stesso modo il lanciatore osservando la matrice di payoff sa che il portiere ha interesse solo a lanciarsi a sinistra, quindi piuttosto che perdere, tirando a sinistra, questo sceglierà sempre (D).

Questo procedimento si traduce nel principio di equilibrio per le strategie pure per dominanza: ogni giocatore sceglie in maniera univoca una mossa (strategia pura), in modo da cercare massimizzare il guadagno e limitare le perdite dopo aver scartato le strategie oggettivamente peggiori.

Al contrario, in situazioni di gioco simili alla prima matrice proposta, i giocatori possono solo assegnare ad ogni mossa una probabilità con la quale giocarla.

**Definizione 3.1.** Si chiama strategia del giocatore  $i$  una funzione  $\sigma_i$  che assegna al giocatore  $i$  una mossa per ogni possibile situazione di gioco. Una  $n$ -upla  $(\sigma_1, \dots, \sigma_N) \in \prod_{k=1}^N$  è detta profilo di strategie.

Una strategia di un giocatore può distinguersi in

- *strategia pura*: assegna in modo univoco una mossa per ogni situazione di gioco; ciò comporta che in condizioni di gioco uguali tra loro, il decisore compie sempre la stessa mossa;
- *strategia mista*: per ogni situazione di gioco si assegna una distribuzione di probabilità sull'insieme delle mosse possibili.

*Osservazione 10.* Nel caso di gioco ripetuto  $G$  con matrice dei payoff  $A$  (ovvero uno schema nel quale i due giocatori si trovano a dover ripetere sempre lo stesso gioco), un giocatore che adotta una strategia pura ad ogni giocata gioca sempre la stessa mossa, mentre un giocatore che adotta una strategia mista sceglie una mossa per ogni round secondo la distribuzione di probabilità della strategia.

### 3.1.1 Equilibrio di strategie

Nel caso di strategie pure, ci si chiede se dato un gioco con matrice di payoff  $A$ , esista un profilo strategico puro di equilibrio, ovvero se una volta arrivati a giocare le mosse  $(i^*, j^*)$  entrambi i giocatori non hanno interesse ad essere gli unici a cambiare strategia.

Nel caso dell'esempio 3, per la seconda matrice proposta la strategia d'equilibrio consiste nel profilo strategico puro (D,S). Un profilo di questo genere non è detto che esista per una matrice di gioco  $A$  qualsiasi, mentre si ha che

**Proposizione 3.2.** *il gioco ammette un equilibrio se solo se*

$$\max_i \min_j a_{i,j} = \min_j \max_i a_{i,j} \quad (3.4)$$

*e la strategia corrispondente si dice strategia di minimax.*



In sostanza il profilo strategico di minimax corrisponde al caso in cui entrambi i giocatori vogliono massimizzare la loro utilità nell' ipotesi che l' avversario faccia la scelta più giusta.

Per quanto riguarda le strategie miste, si introduce la funzione di *payoff atteso*:

$$E(x, y) = \sum_{i=1}^m \sum_{j=1}^n a_{i,j} x_i y_j = x^T A y \quad (3.5)$$

dove  $x = (x_1, \dots, x_m)$ ,  $\sum_{i=1}^m x_i = 1$ , è una strategia mista per il giocatore 1 e  $y = (y_1, \dots, y_n)$ ,  $\sum_{i=1}^m y_i = 1$  è una strategia mista per il giocatore 2. La condizione di equilibrio delle strategie miste rispecchia il principio espresso per le strategie pure, ovvero è una coppia di strategie miste che fa sì che a nessuno dei due giocatori convenga cambiare strategia nell' ipotesi che l' avversario non la cambi.

In termini di payoff atteso, una strategia  $(\bar{x}, \bar{y})$  è di equilibrio se

$$E(x, \bar{y}) \leq E(\bar{x}, \bar{y}) \leq E(\bar{x}, y) \quad (3.6)$$

per ogni  $x$  strategia mista del giocatore 1 e  $y$  strategia mista del giocatore 2. Al contrario del caso precedente, von Neumann nel 1928 ha dimostrato che

**Teorema 3.3** (Von Neumann). *Ogni gioco a somma zero tra due giocatori ammette un profilo strategico misto d' equilibrio.*

Nel 1951, Nash pubblicò un risultato più generale

**Teorema 3.4.** *Ogni gioco tra un numero finito di giocatori, ciascuno dei quali dispone di un numero finito di possibili mosse, ammette un equilibrio.*

Nonostante il risultato esistenziale, tuttavia non esiste ancora un modo per calcolare con esattezza un tale equilibrio, detto *equilibrio di Nash*, in tempo polinomiale.

### 3.1.2 Giochi con mancanza di informazioni.

La teoria standard dei giochi, fino al lavoro di Harsanyi del 1968, assumeva che le caratteristiche del gioco fossero conoscenza comune dei giocatori, nello specifico i giocatori sanno quale sono le matrici di payoff ad ogni giocata, le possibili strategie degli avversari e le funzioni di utilità (le preferenze di ogni giocatore a riguardo dello scopo del gioco).

Tuttavia in molte circostanze reali i giocatori non sono a conoscenza di tutti questi elementi; in questo caso si parla di *giochi a conoscenza incompleta*.

Un classico esempio di gioco ad informazione incompleta consiste nei *giochi bayesiani* introdotti da Harsanyi; ad esempio:

*Esempio 4.* Si consideri un gioco a due giocatori ( I e II ) nel quale tuttavia il giocatore I può assumere due "tipi" diversi, ad esempio può decidere se tirare la palla con *potenza* (giocatore  $I_P$  ) oppure prediligere l' *effetto* (giocatore  $I_E$ ). Supponiamo che a seconda del tipo di giocatore ci sia una matrice di gioco diversa, quindi  $A_P$  e  $A_E$ .

A questo punto supponiamo che il tiratore scelga a caso se essere un giocatore di potenza o d'effetto. Ovviamente non dirà la scelta al portiere.

Più in generale si può pensare che valga un principio uguale per il giocatore II, tuttavia per semplicità consideriamo che II sia di un solo tipo.

Una situazione del genere è stata approcciata da Harsanyi includendo un terzo giocatore nella partita, la *Natura*, la cui mossa consiste nel selezionare un gioco tra quelli possibili (ovvero una matrice di payoff), e in questo caso particolare solo il giocatore I è a conoscenza della mossa effettuata da questo ente.

Considerato un gioco con la caratteristica appena introdotta, ovvero ogni giocatore può avere più di un tipo e giocare di conseguenza, sarà fondamentale per ogni giocatore una stima a riguardo del tipo di avversario. Ovvero un giocatore (nel caso dell' esempio4 il giocatore II) dovrà attribuire una probabilità per ogni tipo di avversario, e di conseguenza al gioco che si sta giocando.

Più formalmente un gioco bayesiano a due giocatori può essere formalizzato

come una quintupla

$$G^b = (\{1, 2\}, \{C_1, C_2\}, \{T_1, T_2\}, \{p_1, p_2\}, \{u_1, u_2\}) \quad (3.7)$$

dove

- $\{1, 2\}$  è l'insieme dei giocatori;
- $C_i$  è l'insieme delle mosse possibili per il giocatore  $i$ ;
- $T_i$  è l'insieme dei tipi possibili di giocatore  $i$ ;
- $p_i$  è una distribuzione di probabilità, indica la probabilità che il giocatore  $i$  assegna ad ogni tipo di avversario;
- $u_i : C_1 \times C_2 \times T_1 \times T_2 \rightarrow \mathbb{R}$  è la funzione di utilità del giocatore  $i$ , ovvero una funzione che in base alle mosse giocate e ai tipi di giocatori (quindi alla matrice payoff corrispondente), associa un numero. Questo numero è un indice corrispondente alla volontà del giocatore in questione. Negli esempi illustrati precedentemente, possiamo considerare come funzione di utilità semplicemente il payoff della giocata per i due giocatori (quindi l'obiettivo finale sarà quello di massimizzare il payoff).

La distribuzione di probabilità che un giocatore assegna ai tipi degli altri giocatori è chiamata *belief*. Questa quantità ovviamente ha un ruolo centrale nello studio dei giochi bayesiani, in quanto come già introdotto, ad ogni tipo di avversario corrisponde un gioco diverso; avere una buona *belief*, o una buona approssimazione del tipo di avversario, comporta ovviamente dei grossi vantaggi sulla scelta di un piano d'azione più efficace.



## Capitolo 4

# Applicazione di Modelli di Markov nascosti ai Giochi Markoviani.

Un gioco di Markov può essere descritto come un gioco ripetuto nel quale la matrice di payoff cambia ogni round seguendo secondo uno schema di catena di Markov.

Qualora nel gioco uno o più giocatori non siano a conoscenza di tutte le informazioni necessarie per definire una strategia opportuna, come le mosse dell' avversario o la matrice dei payoff, si dice che il gioco è a informazione incompleta.

In questo capitolo si cercherà di risolvere il problema introdotto da Renault in [11] e studiato da Bressaud in [12] e Horner in [13], tuttavia si cercherà un approccio pratico per lo studio del problema, andando oltre l' intenzione di trovare strategie ottimali di equilibrio. Si cercherà di implementare un metodo pratico per giocare nei panni di un giocatore non informato riguardo lo stato attuale del sistema, ma a conoscenza delle possibili matrici di payoff e supponendo che l' alternanza dei giochi sia una catena di Markov omogenea. Osservando le azioni del giocatore informato dello stato attuale del sistema, si cercherà un modo per tentare di anticipare le mosse future e di conseguen-

### **38. Applicazione di Modelli di Markov nascosti ai Giochi Markoviani.**

---

za avere un guadagno.

Questa sezione è ispirata parzialmente al lavoro di Benevides [14], nel quale l' autore suppone che il giocatore informato giochi a seconda dello stato attuale, del quale solo lui è a conoscenza, con la strategia di equilibrio di Nash mista riguardante il gioco specifico. Il giocatore non informato, quindi, ha il compito di ricostruire la catena di Markov sottostante tramite algoritmo di Baum-Welch ad ogni giocata.

Benevides ha dimostrato sperimentalmente la validità di questo procedimento, in confronto ad altri metodi precedentemente utilizzati.

Nel gioco introdotto da Renault, la situazione è capovolta, in quanto il giocatore non informato ha a disposizione la conoscenza della matrice di transizione della catena sottostante, ma la situazione di conflitto è creata dal fatto che a priori non si può supporre che il giocatore informato giochi solo in base allo stato attuale. Si introdurrà il problema e si daranno dei risultati teorici senza dimostrazione; per eventuali chiarimenti si rimanda alla lettura di [11], [12], [13].

#### **4.1 Gioco di Renault e strategie ottimali.**

Si introduce il gioco di Renault, ovvero un gioco ripetuto a due giocatori, a somma zero e ad informazione incompleta con le seguenti caratteristiche:

- il giocatore1 è informato riguardo la matrice di payoff attuale del gioco, mentre il giocatore2 ne è all' oscuro;
- ad ogni istante il gioco può assumere una delle due seguenti matrici di payoff per il giocatore1

$$\bar{s} = \begin{vmatrix} 1 & 0 \\ 0 & 0 \end{vmatrix} \quad \underline{s} = \begin{vmatrix} 0 & 0 \\ 0 & 1 \end{vmatrix} \quad (4.1)$$

si supporrà tuttavia che il giocatore2 conosca le matrici  $\bar{s}$  e  $\underline{s}$  e il fatto che non ve ne siano altre;

- ad ogni giocata la matrice dei payoff cambia seguendo lo schema di catena di Markov simmetrica, ovvero con probabilità  $p$  il gioco rimane lo stesso, mentre con probabilità  $1 - p$  si assisterà al cambio di gioco nel turno successivo. Si suppone che entrambi i giocatori siano al corrente sia della natura del problema, sia del parametro  $p$  che regola il processo;
- alla fine di ogni turno di gioco, entrambi i giocatori osservano la mossa dell' avversario, ma solo il giocatore informato dello stato attuale può risalire al payoff della giocata, mentre tutto ciò rimane ignoto al secondo.
- si suppone che il giocatore1 scelga ad ogni turno la riga della matrice, chiamando con  $T$  (Top) la prima riga, e con  $B$  (Bottom), la seconda, mentre il giocatore2 dovrà scegliere tra la prima colonna, indicata con  $L$  (Left) e la seconda, indicata con  $R$  (Right).

La particolarità di questo gioco risiede nella particolare semplicità delle matrici di payoff: ovviamente se entrambi i giocatori avessero la conoscenza del gioco in vigore ad ogni passo, la loro strategia sarebbe banalmente descrivibile con

1. se il gioco è  $\bar{s}$  il profilo strategico di equilibrio è formato dal profilo di strategia pura  $(1, 2)$ , ovvero il giocatore1 sceglierà sempre la prima riga, mentre il giocatore2 sempre la seconda colonna;
2. viceversa in  $\underline{s}$  il giocatore1 non ha mai soluzioni migliori della seconda riga, mentre il giocatore2 sceglie sempre la prima colonna: si osservano le mosse  $(2, 1)$ .

Il problema si complica per il fatto che il giocatore2 non sa mai quale è la matrice dei payoff; ciò permette al giocatore1 di avere dei turni di gioco in cui riesce a centrare l' obiettivo di incrementare la propria vincita.

Tuttavia il giocatore1 non può ignorare che l'avversario conosce la probabilità che nel turno successivo il gioco rimanga lo stesso, quindi dovrà agire di conseguenza: ad esempio nel caso limite  $p = 1$ , ovvero nell' ipotesi che il

#### 40. Applicazione di Modelli di Markov nascosti ai Giochi Markoviani.

gioco rimarrà sempre lo stesso, se il giocatore informato punterà troppo a massimizzare la propria vincita rischierà di informare l' avversario, che di conseguenza non avrà difficoltà nell' annullare i successivi payoff di 1.

Viceversa, il caso  $p = 0.5$  è completamente favorevole al giocatore informato: può giocare per massimizzare il guadagno in ogni turno di gioco, conscio del fatto che la mossa giocata non può dare informazione alcuna al giocatore2.

Nei casi intermedi, considerando  $p \in [0.5, 1]$ , il giocatore informato dovrà cercare di mediare tra scelte improntate sul profitto immediato e mosse che sa di ritenere "errate", ma che sono utili a mandare in confusione il suo avversario.

Si osserva che è possibile studiare il problema considerando il parametro  $p$  solo nell' intervallo  $[0.5, 1]$ , in quanto per valori più piccoli  $c'$  è simmetria. Ad esempio il caso  $p = 0$  è speculare al caso  $p = 1$ : nel primo caso il gioco in ogni turno sarà diverso dal gioco nel turno precedente, quindi se il giocatore non informato riesce a stabilire con esattezza il gioco in un turno, riuscirà a garantire la perdita dell' avversario in ogni turno successivo, alternando le proprie mosse tra  $L$  e  $R$ ; nel caso  $p = 1$  il gioco in ogni turno resta quello del gioco precedente, quindi il giocatore non informato riesce ad annullare i guadagni dell' avversario, scegliendo opportunamente sempre la stessa mossa (nel caso il gioco sia  $\bar{s}$  il giocatore deve giocare sempre  $R$  e viceversa).

Per quanto riguarda i casi intermedi, si ha come specularità per i casi  $p_1 = 0.5 - \delta$  e  $p_2 = 0.5 + \delta$ , con  $\delta \in [0, 0.5]$ : ad esempio considerando  $p_1 = 0.4$  e  $p_2 = 0.6$  si ha che i giocatori sanno che

- il gioco ad ogni turno cambia con probabilità del 40% nel caso  $p = 0.4$ ;
- in ogni round il gioco è lo stesso di quello precedente con probabilità del 40% nel caso  $p = 0.6$ .

Quindi le scelte dei giocatori, e le informazioni ottenute dopo ogni giocata, sono confrontabili da un punto di vista qualitativo nei due casi speculari; piuttosto che il valore del parametro di transizione  $p$ , il gioco è influenzato dalla vicinanza o meno del parametro ai valori estremi: se  $p$  è vicino a 0 o a 1,



allora il giocatore informato deve giocare le proprie mosse in modo quasi del tutto casuale, viceversa l' avversario potrebbe inferire a riguardo del gioco in modo corretto; per valori di  $p$  man mano più vicini a 0.5, al contrario, il giocatore non informato osservando le mosse dell' avversario può inferire meno a riguardo del gioco in ogni turno.

Quindi il gioco si presenta come un gioco di Markov con mancanza di informazione per un concorrente come già introdotti nel capitolo precedente, associando le due matrici di  $\bar{s}$  e  $\underline{s}$  come *tipi* per il giocatore1.

A questo punto la *belief* del giocatore2 giocherà un ruolo di fondamentale importanza: essa è semplicemente la probabilità che il giocatore non informato assegna ai due giochi: se ad esempio il giocatore2 ha una belief sbilanciata su  $\bar{s}$  ci si aspetta che giochi di conseguenza, ovvero che scelga la seconda colonna.

#### 4.1.1 Strategie di equilibrio.

Nel suo lavoro, Renault [11] ha provato che, considerato il gioco introdotto nella sezione precedente, e ripetuto infinite volte, allora questo ha *valore*, ovvero il limite per il numero di round infinito della massima vincita attesa del giocatore informato, per  $p \in [1/2, 2/3]$  e per alcuni parametri isolati tra cui  $p^* \simeq 0.7589$ .

Bressaud e Horner, rispettivamente nei lavori [12] - [13], hanno ripreso il lavoro di Renault, e hanno fornito, per uno schema a infiniti turni del gioco precedentemente introdotto, delle strategie miste ottimali per i due giocatori. Il lavoro si basa sullo studio di *Processi Decisionali di Markov Parzialmente Osservabili* (POMDP) a orizzonte infinito; è da specificare che le strategie sono dimostrate essere ottimali solo per  $p \in [1/2, 2/3]$  e per  $p^*$ , mentre si è dimostrato che non lo sono per alcuni parametri fuori da  $[1/2, 2/3]$ .

Il problema per i  $p$  qualsiasi è tutt' ora aperto, tuttavia riportiamo le strategie ottimali per i  $p \in [1/2, 2/3]$  per il giocatore1 e giocatore2 proposte da Bressaud:

## 4.2. Applicazione di Modelli di Markov nascosti ai Giochi Markoviani.

1. per il giocatore1 si è dimostrato essere ottimale una strategia mista  $\sigma^*$  che gioca sulla *belief* del suo avversario: si considera il parametro  $\theta$ , che ricorsivamente indica la *belief* del giocatore2 nel caso quest' ultimo assuma che il giocatore informato usi la strategia  $\sigma^*$ . Si definiscono le funzioni

$$f_B(\theta) = \begin{cases} p \frac{2\theta-1}{\theta} + (1-p) \frac{1-\theta}{\theta}, & \text{se } \theta \geq \frac{1}{2}, \\ 1-p, & \text{se } \theta \leq \frac{1}{2} \end{cases} \quad (4.2)$$

$$f_T(\theta) = \begin{cases} p, & \text{se } \theta \geq \frac{1}{2} \\ p \frac{\theta}{1-\theta} + (1-p) \frac{1-2\theta}{1-\theta}, & \text{se } \theta \leq \frac{1}{2} \end{cases} \quad (4.3)$$

A questo punto la strategia  $\sigma^*$  è definita come segue: il giocatore sceglie  $T$  (prima riga) con probabilità

$$\mathbb{P}(\text{giocare } T) = \begin{cases} 1 & \text{se il sistema è in } \bar{s} \text{ e } \theta \leq \frac{1}{2}; \\ \frac{1-2\theta}{1-\theta} & \text{se il sistema è in } \underline{s} \text{ e } \theta \leq \frac{1}{2}; \\ \frac{1-\theta}{\theta} & \text{se il sistema è in } \bar{s} \text{ e } \theta \geq \frac{1}{2}; \\ 0 & \text{se il sistema è in } \underline{s} \text{ e } \theta \geq \frac{1}{2}; \end{cases} \quad (4.4)$$

Altrimenti gioca  $B$ . Ad ogni nuova giocata, il valore di  $\theta$  si aggiorna

$$\theta_1 = \begin{cases} f_T(\theta) & \text{se ha giocato } T; \\ f_B(\theta) & \text{se ha giocato } B. \end{cases} \quad (4.5)$$

2. per il giocatore2, si dimostra che la strategia d' equilibrio, è basata sull' ultima mossa del giocatore informato: se il giocatore1 gioca  $T$ , allora nel turno successivo si giocherà

$$\begin{cases} R & \text{con probabilità } \frac{2p}{4p-1}; \\ L & \text{con probabilità } 1 - \frac{2p}{4p-1} \end{cases} \quad (4.6)$$

Se l' avversario avesse scelto  $B$  le probabilità di giocata sarebbero invertite.

*Osservazione 11.* Si nota che la strategia di equilibrio per il giocatore informato si basa esclusivamente sulle mosse da esso compiute e sullo stato attuale del sistema, mentre è svincolata dalle mosse dell' avversario non informato. Viceversa le mosse del giocatore2 sono dettate esclusivamente dall' ultima giocata dell' avversario.

## 4.2 Strategie basate su Modelli di Markov Nascosti.

Osservando le strategie di equilibrio introdotte nella sezione precedente si ha che

- le mosse del giocatore1 ad ogni turno sono aleatorie ma dipendono dalle proprie mosse passate e dal parametro  $p$  della matrice di transizione;
- le mosse del giocatore2 ad ogni turno sono aleatorie ma dipendono dalla mossa precedente del giocatore1 e da  $p$ .

In particolare quindi in questo caso le mosse del giocatore non informato non dipendono ad ogni turno da tutte le mosse del proprio avversario, ma solo dall' ultima osservata.

Si introduce a questo punto un altro approccio di giocata per il giocatore non informato, basato sulle catene di Markov nascoste: dal punto di vista del giocatore2 infatti si hanno

- una catena di Markov non osservabile con matrice di transizione  $P$ , con spazio degli stati  $S = \{\bar{s}, \underline{s}\}$ , che ad ogni turno indica il gioco corrente;
- le mosse dell' avversario, ovvero la scelta della riga della matrice di payoff, osservate nei turni precedenti, quindi dei segnali nello spazio  $V = \{T, B\}$ .

#### 44. Applicazione di Modelli di Markov nascosti ai Giochi Markoviani.

E' bene osservare che le mosse del giocatore1, quindi i segnali emessi, in generale non dipendono solo dallo stato (gioco) attuale, ma possono dipendere anche dalle mosse precedenti (come nella strategia ottimale). Tuttavia prendiamo in considerazione un giocatore non informato che pensi che le scelte dell' avversario in realtà dipendano solo dal gioco attuale.

Sotto queste ipotesi, si può facilmente formalizzare la giocata del giocatore informato come uno schema di Markov Nascosto.

In questo caso, dato che anche il giocatore non informato conosce la matrice di transizione  $P$  del sistema, rimane l' incognita della matrice di emissione  $2 \times 2$ , che chiameremo  $b_{vera}$ , e quindi in particolare della conoscenza delle reali probabilità con le quali il giocatore informato gioca per ottenere il guadagno ad ogni turno ( $b_{vera}(1, 1)$  e  $b_{vera}(2, 2)$  rispettivamente per  $\bar{s}$  e  $\underline{s}$ ) o viceversa le probabilità con le quali il giocatore1 gioca per confondere le idee del giocatore2.

A tal scopo si considererà inizialmente una matrice di emissione inizializzata dal giocatore2, chiamata  $b_{guess}$ , che verrà man mano aggiornata fruttando gli algoritmi di learning introdotti nei capitoli precedenti, ottenendo delle approssimazioni, chiamate nel proseguio  $b$ , della matrice  $b_{vera}$ .

Ad ogni turno si avrà che il giocatore non informato ha a disposizione un modello di Markov nascosto  $\lambda = (\pi, P, b)$ . Utilizzando tale modello egli può calcolare la probabilità che, date le mosse dell' avversario informato fino a quel punto, nel turno successivo il gioco sia  $\bar{s}$ : indicate con  $\mathbf{m} = (m_1, \dots, m_n)$  le mosse del giocatore1 osservate nei primi  $n$  turni di gioco, si ha

$$\begin{aligned} \theta &:= \mathbb{P}(X_{n+1} = \bar{s} \mid \mathbf{M} = \mathbf{m}) = P_{1,1} \mathbb{P}(X_n = \bar{s} \mid \mathbf{m}) + P_{2,1} \mathbb{P}(X_n = \underline{s} \mid \mathbf{m}) = \\ &= P_{1,1} \frac{\alpha_n(\bar{s})}{\sum_i \alpha_n(i)} + P_{2,1} \frac{\alpha_n(\underline{s})}{\sum_i \alpha_n(i)} \end{aligned} \quad (4.7)$$

dove le quantità  $\alpha$  sono le quantità *forward* descritte da (2.8).

La quantità  $\theta$  è quindi la probabilità che il giocatore2 assegna al gioco  $\bar{s}$  per il prossimo turno, tramite la quale dovrà scegliere tra  $L$  ed  $R$ , rispettivamente prima e seconda colonna della matrice dei payoff, in modo da minimizzare la probabilità di vincita dell' avversario:

- se gioca  $L$  si assume che giocatore1 vinca con probabilità  $\theta \cdot b(1, 1)$ ;
- se gioca  $R$  si lascia al giocatore1 una probabilità di vincita di  $(1 - \theta) \cdot b(2, 2)$ ;

quindi il giocatore2 gioca

$$\begin{cases} L & \text{se } \theta b(1, 1) \leq (1 - \theta) b(2, 2); \\ R & \text{altrimenti} \end{cases} \quad (4.8)$$

*Osservazione 12.* Nel caso di  $b$  simmetrica, ovvero qualora si ritenga che la probabilità di "sbagliare" da parte del giocatore1 non dipenda dallo stato, si ha che il giocatore2 sceglie di giocare la prima colonna se  $\theta \leq \frac{1}{2}$ .

Al contrario, se ad esempio  $b(1, 1)$  è molto minore di  $b(2, 2)$ , il giocatore2 può scegliere di giocare la prima colonna anche se  $\theta > 0.5$ , in quanto anche se il gioco dovesse effettivamente essere  $\bar{s}$ , il giocatore1 ha comunque una buona probabilità di sbagliare (giocando la seconda riga).

Ricapitolando, una partita giocata utilizzando una strategia basata su catene di Markov nascoste può riassumersi nei seguenti passaggi:

1. inizializzare i parametri della catena di Markov nascosta:  $\lambda = (P, b, \pi)$ , dove la matrice di transizione  $P$  è data, la matrice  $b$  viene inizializzata in qualche modo (ad esempio random), e il vettore di probabilità iniziale  $\pi$  inizializzato anch'esso random (oppure  $\pi = (0.5, 0.5)$ );
2. si calcola il valore  $\theta$  secondo la formula (4.2), nel primo turno di gioco si può assegnare  $\theta = \pi_1$ ;
3. si effettua la giocata secondo lo schema (4.8);
4. si osserva la mossa dell'avversario e si aggiunge al vettore delle mosse passate  $\mathbf{M}$ ;
5. si procede all'addestramento della matrice  $b$ , ad esempio tramite formula di ristima di Baum-Welch;
6. si torna al punto 2.

### **4.3 Applicazione del metodo contro un giocatore stazionario.**

Si testa la validità del metodo descritto nel caso in cui il giocatore informato effettivamente sia *stazionario*, ovvero che le probabilità con cui decide tra  $T$  e  $B$  dipendano solo dallo stato del sistema: in questo caso esiste una matrice di emissione "vera", ovvero quella effettivamente utilizzata dal giocatore per generare le proprie mosse.

Per quanto riguarda l'addestramento della matrice di emissione  $b$ , usata dal giocatore<sup>2</sup>, si useranno ad ogni turno le formule di ristima di Baum-Welch (2.45) solo sulla matrice di emissione (avendo già a disposizione la matrice di transizione "vera"). L' unica accortezza per quanto riguarda questo punto è data dal fatto che, come è noto, l' algoritmo tende a massimi locali della verosimiglianza; quindi sono da evitare

- stime iniziali troppo lontane dai valori reali;
- utilizzo dell' addestramento su set di valori molto ridotti: in questo caso la ristima dei parametri porta a matrici  $b$  del tipo

$$b = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (4.9)$$

ma una volta arrivata a forme del genere, ogni eventuale ristima della matrice (allargando il vettore delle osservazioni), rimane della stessa forma. Ciò ovviamente porta all' inefficacia del metodo.

Per sopperire al secondo problema si può procedere come segue:

1. quando la matrice assume forme deterministiche (una entrata assume il valore 1), si perturba la riga che contiene il valore in questione, in modo da permettere alle ristime successive di avere luogo in modo efficace;
2. si comincia il learning della matrice di emissione solo dopo un numero sufficientemente alto di turni; ciò può essere tollerato in prospettiva di partite molto lunghe.

3. ciclicamente la matrice di emissione inferita tramite gli algoritmi di learning può essere reinizializzata con una matrice  $b_{guess}$ ;

Si riportano i risultati di giocate contro un giocatore informato stazionario, confrontando la strategia appena descritta con la strategia ottimale (4.6). Si sono considerate partite da 5000 turni di gioco ciascuna, con stima iniziale della matrice  $b$  di quasi incertezza, ovvero

$$b_{guess} = \begin{bmatrix} 0.55 & 0.45 \\ 0.45 & 0.55 \end{bmatrix} \quad (4.10)$$

e inoltre si sceglie di cominciare ad inferire sulla matrice di emissione dopo 100 turni di gioco.

Si premette che la scelta di  $b_{guess}$  non è stata fatta di assoluta incertezza (tutti gli elementi uguali a 0.5), perchè durante l' inferenza la stima potrebbe sbilanciarsi verso valori opposti alla matrice di emissione reale del giocatore1. In primo luogo si analizzano i risultati delle giocate per il parametro  $p = 0.66$ , quindi nel caso in cui la strategia ottimale presentata per il giocatore2 è dimostrata essere d' equilibrio, al variare di diverse matrici di emissioni del giocatore1:

- caso di strategie sempre dominate: ovvero il giocatore1 decide di scegliere sempre la mossa che lo potrebbe far vincere, ovvero

$$b_{vero} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (4.11)$$

Si sono simulate 70 partite da 5000 turni ciascuna; ovviamente più il metodo sarà efficace, minore sarà il guadagno del giocatore1.

A tal scopo riportiamo i grafici delle percentuali di vincita del giocatore1, ovvero per ogni partita si considera la quantità

$$\text{percentuale} = \frac{\text{guadagno totale}}{\text{numero dei turni di gioco}} \quad (4.12)$$

Come si può osservare nella Figura 4.1, in questo caso la strategia da noi introdotta permette al giocatore informato di guadagnare molto

#### 4.8. Applicazione di Modelli di Markov nascosti ai Giochi Markoviani.

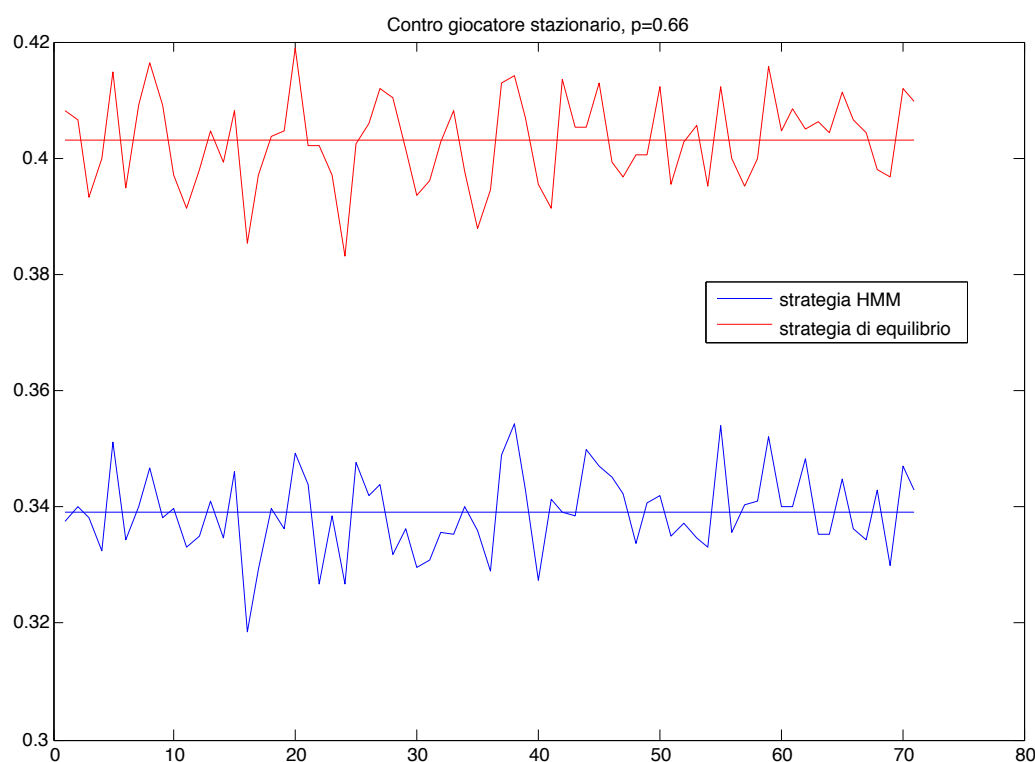


Figura 4.1: Confronto tra le percentuali di vincita del giocatore1 nel caso in cui il giocatore non informato adotti una strategia basata su catene di Markov nascoste (in blu), o quella di equilibrio teorico (rosso).

meno rispetto al giocatore non informato che gioca secondo la strategia introdotta da Bressaud ed Horner: nel primo caso il giocatore informato vince in media con una frequenza di circa 34%, mentre l'adozione della strategia d'equilibrio per il giocatore2 fa sì che il giocatore1 abbia una percentuale di vincita media del 40% (i valori medi sono rappresentati dalle linee orizzontali).

Inoltre in ogni partita tramite ristime via formule di Baum-Welch sulla matrice di emissione  $b$ , si è inferito il valore esatto della matrice di emissione, ovvero dopo poco più di 100 di turni di gioco la matrice



usata dal nostro giocatore2 si è stabilizzata intorno a valori

$$b = b_{vero} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (4.13)$$

- Si testa il metodo modificando la matrice reale di emissione del giocatore1, considerandolo ancora stazionario ma ora con matrice di emissione non diagonale, in modo che esso non riveli sempre il gioco appena giocato. Consideriamo

$$b_{vero} = \begin{bmatrix} 0.8 & 0.2 \\ 0.25 & 0.75 \end{bmatrix} \quad (4.14)$$

e ripetiamo lo stesso confronto tra i metodi proposti per il giocatore2 (il nostro basato su catene nascoste e il secondo basato sulle strategie d' equilibrio).

Anche in questo caso si osserva in Figura 4.2 la maggiore efficacia del metodo da noi proposto rispetto alla strategia ottimale. Inoltre dalle medie delle vincite si osserva che per il giocatore1 conveniva, qualora decidesse di effettuare le proprie scelte solo in base allo stato (schema stazionario), optare per la scelta precedente della matrice di emissione

- osservando che il guadagno del giocatore1 è maggiore nel caso in cui le mosse siano sempre giuste, si testa il metodo nel caso in cui la matrice di emissione reale sia vicina alla matrice (4.11):

$$b_{vero} = \begin{bmatrix} 0.95 & 0.05 \\ 0.05 & 0.95 \end{bmatrix} \quad (4.15)$$

In questo caso su 70 partite da 5000 turni si ottengono delle percentuali di vincita come in Figura 4.3

Si osserva che anche in questo caso il giocatore informato vince ancora meno rispetto ad una strategia completamente dominante (matrice (4.11)); inoltre il metodo basato su catene di Markov nascoste risponde anche in questo caso molto meglio della strategia d' equilibrio.

## 50. Applicazione di Modelli di Markov nascosti ai Giochi Markoviani.

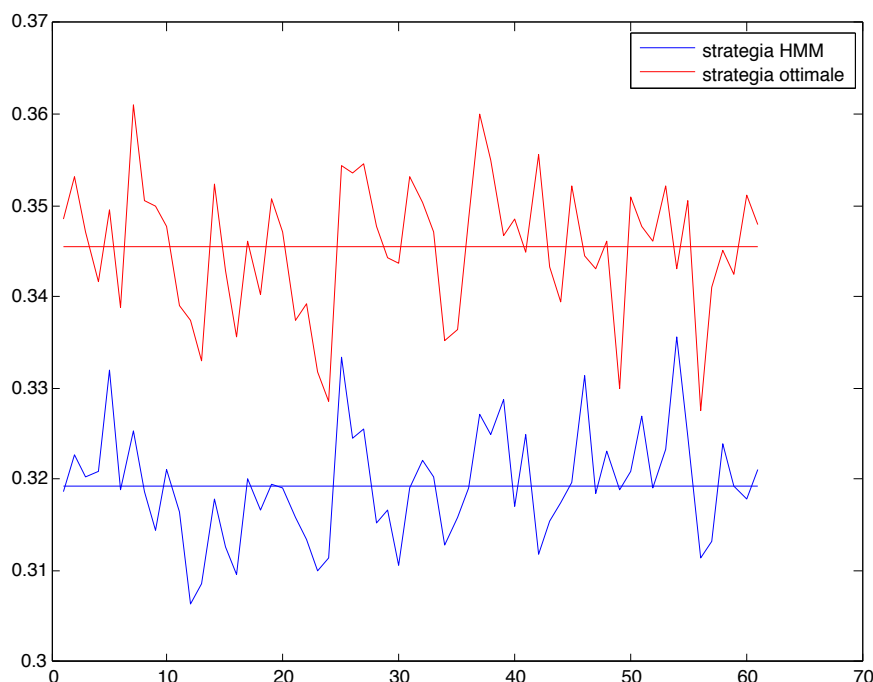


Figura 4.2: Confronto percentuali di vincita di un giocatore informato stazionario di matrice di emissione reale (4.14), nel caso in cui l'avversario adotti la strategia basata sulle catene o la strategia d'equilibrio.

Osservata l'efficacia del nostro metodo nel caso in cui il parametro  $p$  della matrice di transizione

$$P = \begin{bmatrix} p & 1-p \\ 1-p & p \end{bmatrix} \quad (4.16)$$

cada nel range di valori per cui si dimostra l'esistenza delle strategie d'equilibrio per i due giocatori (ossia  $p \in [1/2, 2/3]$ ), si considera una nuova matrice di transizione con parametro  $p$  esterno a tale intervallo; ovvero si riporta l'analisi appena presentata nel caso di

$$P = \begin{bmatrix} 0.8 & 0.2 \\ 0.2 & 0.8 \end{bmatrix} \quad (4.17)$$

ossia prenderemo in esame il caso particolare  $p = 0.8$ .

Si considera inizialmente un giocatore informato stazionario che gioca una

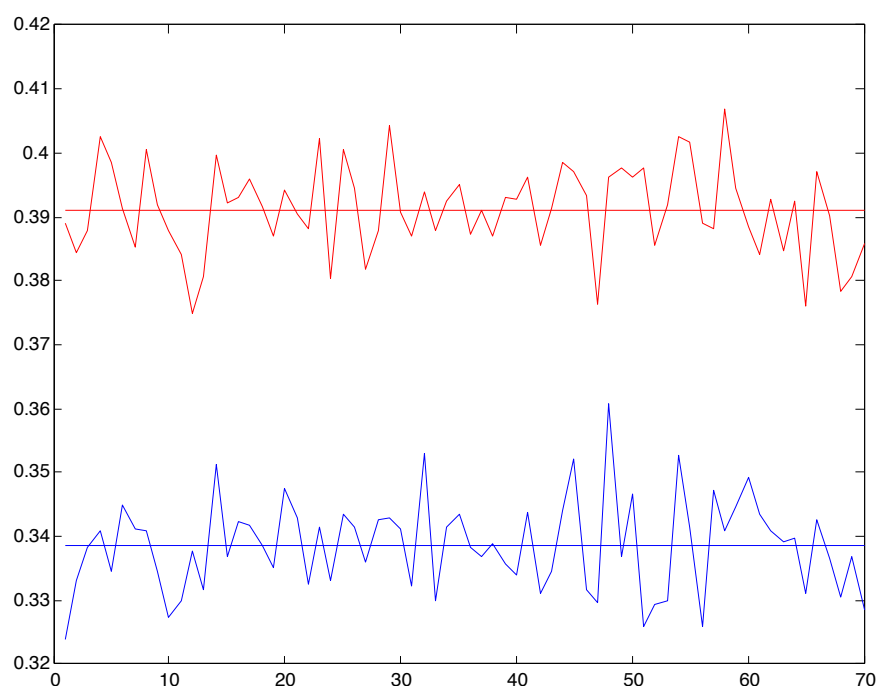


Figura 4.3: Percentuali di vincita nel caso in cui la matrice di emissione vera del giocatore informato sia (4.15). In blu i valori corrispondenti al metodo basato su catene di Markov nascoste, in rosso i valori per strategia d'equilibrio del giocatore non informato.

strategia volta a massimizzare il guadagno in ogni turno di gioco, ovvero un giocatore che gioca a tutti gli effetti secondo uno schema di Markov nascosto con matrice di emissione (4.11). Le percentuali di vincita in 70 partite da 5000 turni di gioco sono illustrate in Figura 4.4. Si osserva che

- il metodo basato sulle catene di Markov nascoste permette all'avversario informato un guadagno sempre minore rispetto alla strategia di equilibrio;
- usando la stessa matrice di emissione (4.11), si osservano vincite per il giocatore informato in generale più basse per  $p = 0.8$  che per  $p = 0.66$ . Ciò è ovviamente dovuto al fatto che in questo ultimo caso il giocatore non informato riesce a farsi una idea più precisa del gioco in ogni turno;

## 52. Applicazione di Modelli di Markov nascosti ai Giochi Markoviani.

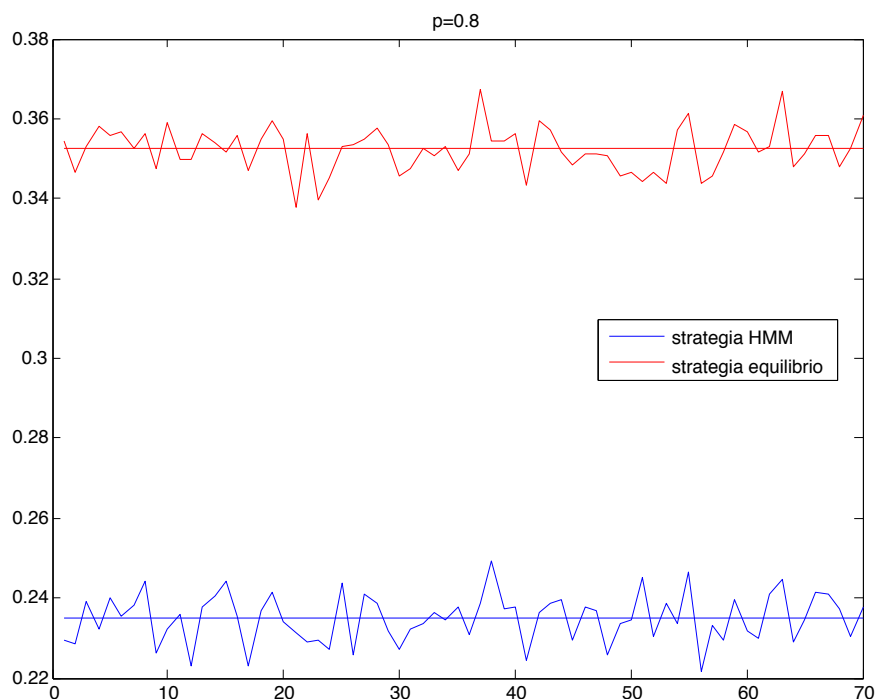


Figura 4.4: Confronto delle percentuali di vincita del giocatore1 a seconda delle strategie del giocatore non informato (in blu lo schema basato su catene nascoste, in rosso la strategia d' equilibrio). Caso di matrice di transizione (4.17) e matrice di emissione effettiva (4.11).

- anche in questo caso il metodo proposto, tramite learning sulla matrice di emissione è in grado di inferire con precisione la reale matrice di emissione, anche partendo da una stima iniziale di incertezza quasi totale, ad esempio la matrice (4.10).

Dato che la transizione tra i giochi è molto più difficile per valori di  $p$  alti, un giocatore che sceglie sempre la mossa giusta gioca una strategia troppo rivelante. Prendiamo in considerazione il caso di matrice di emissione reale

$$b_{vero} = \begin{bmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{bmatrix} \quad (4.18)$$

ovvero che il giocatore informato decida in media 1 volta su 3 di sbagliare la giocata per confondere l' avversario. Si ottengono i risultati della Figura

4.5. Si osserva che, rispetto al caso precedente

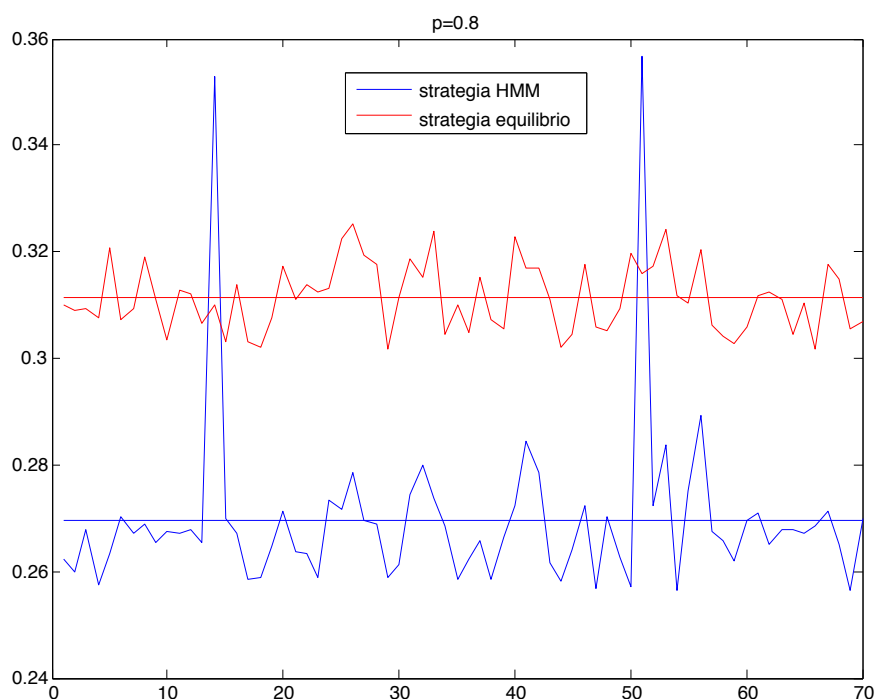


Figura 4.5: Percentuali di vincita del giocatore1, nel caso in cui la matrice di emissione reale sia (4.18)

- contro un giocatore2 con strategia basata su catene nascoste, la scelta di sbagliare in media il 30% delle volte comporta una vincita in media superiore per il giocatore1;
- contro un giocatore con strategia d' equilibrio, sarebbe convenuto al giocatore informato giocare sempre per massimizzare il guadagno ad ogni turno, come nel caso precedente;
- in media per il giocatore non informato conviene usare una strategia basata su processi di Markov nascosti.

Osserviamo che in Figura 4.5 il giocatore con strategia di Markov nascosta può in alcuni casi giocare partite anomale, contraddistinte da un guadagno

## 54. Applicazione di Modelli di Markov nascosti ai Giochi Markoviani.

---

dell' avversario molto superiore alla media (i picchi nel grafico dei guadagni). Ciò è dovuto al fatto che nelle partite in questione, il giocatore2 non è riuscito ad inferire la giusta matrice di emissione dell' avversario; al contrario da un certo turno in poi il giocatore2 ha attribuito all' avversario uno schema di Markov nascosto con matrice di emissione del tipo

$$b = \begin{bmatrix} q & 1 - q \\ q & 1 - q \end{bmatrix} \quad (4.19)$$

in particolare, per questi valori il giocatore non distingue gli stati in base alle mosse, le quali hanno la stessa probabilità di emissione in entrambi i giochi. Questo comporta che la valutazione del parametro  $\theta$ , definito in (4.2), sia in ogni turno

$$\theta = 0.5 \quad (4.20)$$

e quindi i due giochi al turno successivo sono equiprobabili per il giocatore2. Il comportamento appena illustrato si accentua per valori della matrice di emissione vera sempre più vicini alla matrice uniforme. Ad esempio considerando

$$b_{vero} = \begin{bmatrix} 0.6 & 0.4 \\ 0.4 & 0.6 \end{bmatrix} \quad (4.21)$$

si osserva, dalla Figura 4.6 che la presenza di picchi, ovvero valori distanti dalla media, è più alto rispetto al caso precedente; ciò è effettivamente dovuto ad una maggiore frequenza del comportamento illustrato precedentemente.

Un modo possibile per risolvere questo inconveniente nelle giocate può consistere nel resettare i valori di  $b$  nell' eventualità che si osservino forme come (4.19). In particolare per non falsare troppo il modello, dato che fenomeni del genere come osservato sono tipici per matrice di emissione reale vicine alla matrice uniforme, sarà opportuno resettare  $b$  a valori vicini, come (4.10).

Con questa accortezza si ripete il test e si ottengono le percentuali di Figura 4.7.

Si nota che la differenza sostanziale rispetto al problema relativo di Figura 4.6 sta nel fatto che il valore medio della percentuale di vincita del giocato-

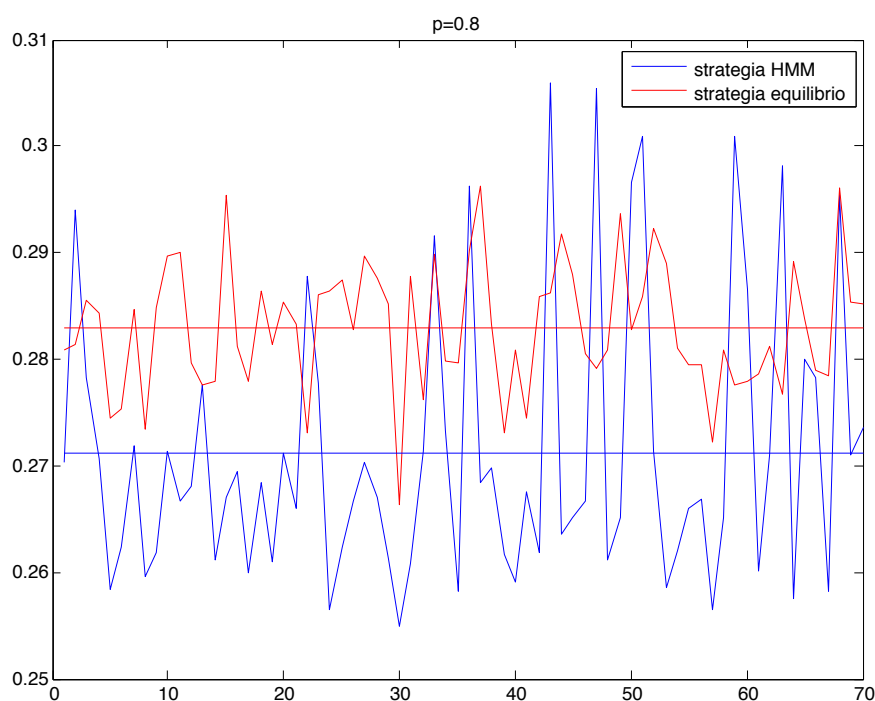


Figura 4.6: Confronto percentuali di guadagno per il giocatore1 in base alla strategia adottata dall' avversario. Caso di giocatore1 stazionario con matrice di emissione reale (4.21). Matrice di transizione tra i giochi con parametro  $p = 0.8$ .

rel, contro il giocatore con strategia basata su catene nascoste, è minore.

Quantitativamente, la piccola modifica ha portato ad un miglioramento medio di circa il 5%, ed inoltre non si osserva più un gran numero di partite anomale (nelle quali il giocatore informato riesce a guadagnare molto più rispetto alla media).

E' buona norma, quindi, effettuare un controllo sulla matrice di transizione usata nel corso di ogni partita.

## 54. Applicazione di Modelli di Markov nascosti ai Giochi Markoviani.

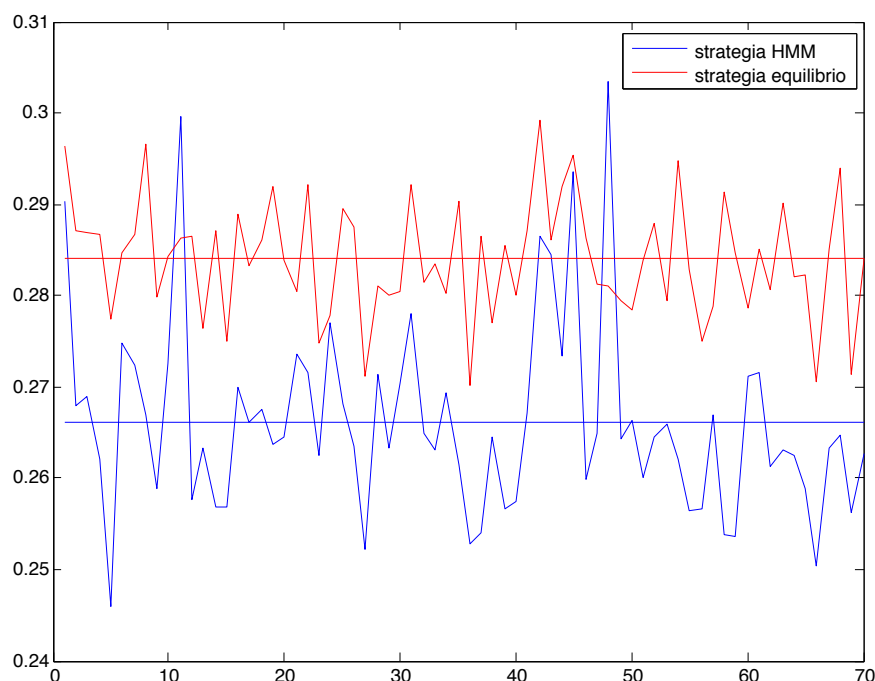


Figura 4.7: Percentuali di vincita del giocatore stazionario con matrice di emissione vera (4.21). In questo caso la strategia basata su catene nascoste è stata modificata per evitare inferenze di matrice di emissione della forma (4.19)

### 4.4 Prova del metodo contro giocatore ottimale.

Dopo aver osservato che il metodo basato sulle catene di Markov nascoste risponde in modo appropriato, migliore della strategia di equilibrio del problema, qualora il giocatore informato giochi effettivamente secondo uno schema di Markov nascosto (quindi la probabilità di giocare  $T$  piuttosto che  $B$  dipendente solo dal gioco attuale), si confrontano le due strategie nel caso il giocatore informato giochi secondo la strategia d'equilibrio descritta dalle (4.4).



Si prende in considerazione la matrice di transizione

$$P = \begin{bmatrix} 0.66 & 0.34 \\ 0.34 & 0.66 \end{bmatrix} \quad (4.22)$$

A questo punto si confrontano i guadagni del giocatore1, giocante strategie di equilibrio, nel caso in cui il giocatore2 giochi

- strategie basate su catene di Markov nascoste (HMM), disegnate in blu, considerando il learning dei parametri solo sulla matrice di emissione  $b$ , e partendo da una stima iniziale

$$b_{guess} = \begin{bmatrix} 0.55 & 0.45 \\ 0.45 & 0.55 \end{bmatrix} \quad (4.23)$$

- strategie di equilibrio descritte da (4.6).

Si confrontano i risultati di 70 partite da 5000 turni ciascuna. Si osserva dalla Figura 4.8 che la strategia da noi proposta permette un guadagno del giocatore1 in media di poco minore rispetto a quello ottenuto mediante strategia ottimale; tuttavia i due metodi rispondono numericamente in modo analogo. Inoltre confrontando quest' ultimo risultato con i grafici in Figura 4.1 e Figura 4.2, relativi alla stessa matrice di transizione per la catena dei giochi, possiamo osservare che

1. al giocatore non informato conviene utilizzare una strategia basata sulle catene nascoste, anche qualora il suo avversario non giochi effettivamente seguendo uno schema stazionario, ma basi le sue scelte sullo storico delle mosse, come ad esempio nella strategia d' equilibrio, in quanto la risposta è nel caso peggiore paragonabile alla strategia ottimale, mentre se il giocatore informato è di tipo stazionario il guadagno dell' uso della strategia basata sulle catene di Markov nascosta è molto conveniente;
2. il giocatore informato trova guadagno nel giocare la strategia d' equilibrio, in alternativa riesce a massimizzare il guadagno solo scegliendo di giocare sempre la mossa per massimizzare il payoff corrente;

## 5.8. Applicazione di Modelli di Markov nascosti ai Giochi Markoviani.

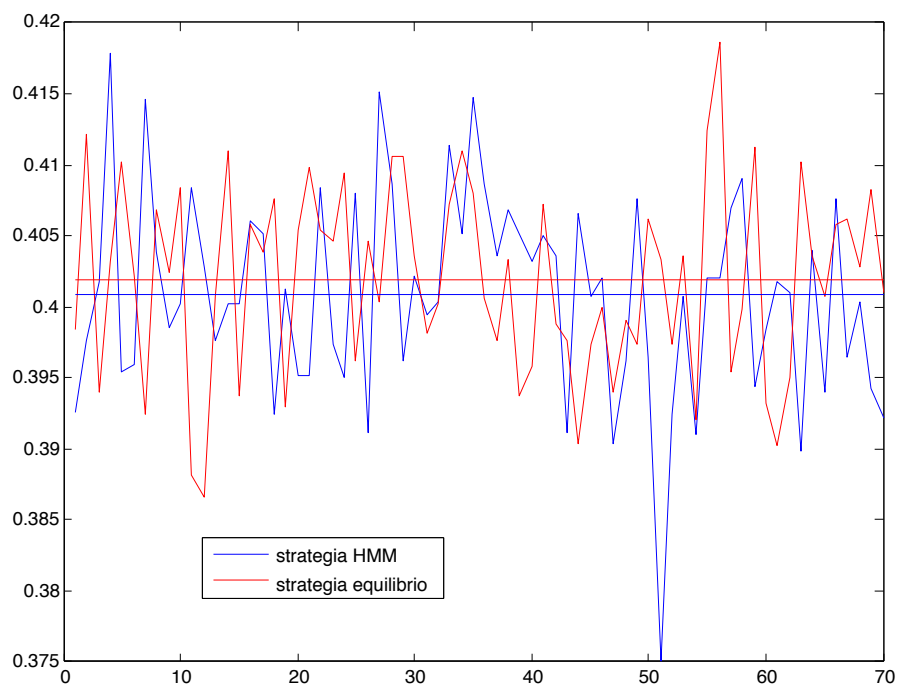


Figura 4.8: Confronto tra vincite del giocatore informato giocante strategia d'equilibrio nel caso l'avversario non informato giochi la rispettiva strategia d'equilibrio (in rosso), oppure strategie basate su catene nascoste (in blu). Le linee orizzontali rappresentano le medie.

3. si osserva che la matrice di emissione  $b$ , nel corso del learning, tende a stabilizzarsi a valori vicini a

$$b = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix} \quad (4.24)$$

ovvero alla lunga il giocatore2 attribuisce uguale probabilità alla mossa del giocatore1; tuttavia bisogna specificare che durante tutto il corso delle partite la matrice di emissione inferita è a colonne costanti, ovvero

$$b(1, 1) = b(2, 1); \quad (4.25)$$

$$b(2, 1) = b(2, 2) \quad (4.26)$$

cioè in questo caso, il giocatore non informato che gioca la strategia secondo schema basato su catene nascoste inferisce che il giocatore in-

formato non gioca in maniera strettamente dipendente dallo stato, o comunque non basa la scelta tra  $T$  o  $B$  sul fatto che il gioco sia  $\bar{s}$  piuttosto che  $\underline{s}$ .

Ciò effettivamente è in parte vero, in quanto come osservato il giocatore che gioca secondo strategie di equilibrio effettua scelte dipendenti sia dallo stato attuale, che dalle proprie mosse passate, in modo da confondere l'avversario.

A prova di ciò è altrettanto notevole il fatto che il giocatore che segue la strategia da noi proposta, contro il giocatore che gioca strategie ottimali, in ogni turno è portato a considerare i giochi  $\bar{s}$  e  $\underline{s}$  come equiprobabili, ovvero in ogni turno:

$$\theta = 0.5 \tag{4.27}$$

In particolare si osservano comportamenti simili ai casi descritti in Figura 4.5 e Figura 4.6 Ripetiamo il test per esaminare le differenze tra le strategie, nel caso di  $p = 0.8$ , quindi fuori dal range di valori per i quali le strategie d'equilibrio sono dimostrate essere effettivamente d'equilibrio. Si ottengono i risultati riassunti in Figura 4.9.

Si nota che anche in questo caso da un punto di vista prettamente numerico, le strategie permettono al giocatore informato guadagni comparabili, sebbene la strategia d'equilibrio in questo caso risponde leggermente meglio in media.

Anche in questo caso la strategia basata su catene di Markov nascoste si stabilizza in ogni partita a condizioni di indifferenza tra gli stati: le matrici di emissione inferite sono del tipo (4.19), e di conseguenza la stima di  $\theta$  risulta essere in ogni turno  $\theta = 0.5$ .

## 60. Applicazione di Modelli di Markov nascosti ai Giochi Markoviani.

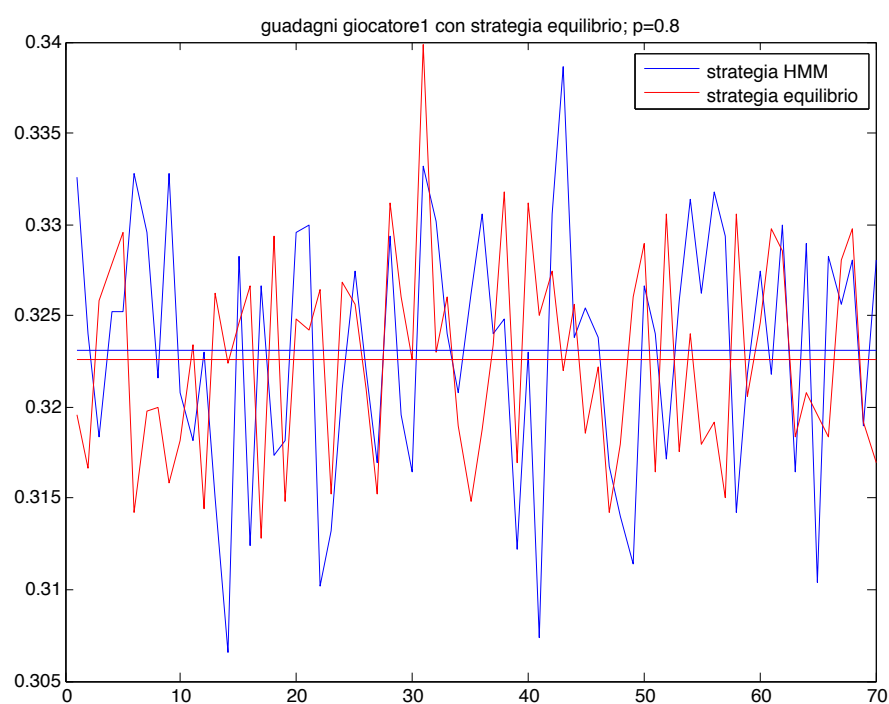


Figura 4.9: Confronto strategie del giocatore non informato nel caso di giocatore1 con strategia d' equilibrio. Matrice di transizione tra stati (4.17).

## 4.5 Gioco permanente per lunghi tratti. Una possibile strategia per il giocatore informato.

Abbiamo esaminato i casi di gioco stocastico di Renault con parametro  $p$  sia nel range  $[\frac{1}{2}, \frac{2}{3}]$ , nel quale sono state trovate strategie di equilibrio, che per  $p = 0.8$ , caso nel quale non è garantito esistano strategie ottimali.

In questo paragrafo si esaminerà il caso di  $p$  ancora più vicino ad 1, ovvero casi in cui il gioco tende ad assumere per lunghi tratti la stessa matrice di payoff.

Nel caso limite  $p = 1$ , per il quale il gioco è sempre o  $\bar{s}$  o  $\underline{s}$ , l' unica scelta sensata per il giocatore1 è quella di giocare mosse assolutamente casuali ad ogni turno, al fine di non permettere all' avversario di inferire la matrice di payoff corretta e conseguentemente annullare ogni profitto futuro. In questo caso, quindi, il guadagno medio percentuale del giocatore1 è del 25%, corrispondente alla probabilità di scelta della riga giusta da parte del giocatore1 (50%) e allo stesso tempo della colonna corrispondente del giocatore2, anch' essa probabile al 50% se quest' ultimo gioca totalmente a caso.

Per valori di  $p$  vicini ad 1, ci si aspetta che il giocatore1 voglia vincere con una percentuale superiore al 25%, senza tuttavia compiere scelte troppo rivelanti riguardo il gioco, in quanto le matrici payoff tendono a permanere per molti turni.

Una possibile scelta da parte del giocatore1 potrebbe consistere nel giocare in modo quasi casuale, e quindi lasciare poche informazioni all' avversario, durante la permanenza del gioco in una data matrice di payoff e giocare la mossa corrispondente al gioco (la mossa che permette di aumentare il guadagno) nel turno in cui egli osserva il cambio di matrice.

Più formalmente, si costruisce una strategia per il giocatore1 che ad ogni turno si basa sul gioco attuale e il gioco precedente in modo che

- se il gioco attuale è lo stesso di quello al turno precedente, si gioca la

## 62. Applicazione di Modelli di Markov nascosti ai Giochi Markoviani.

mossa per arrivare al guadagno con probabilità  $\tau$ ;

- se il gioco attuale è diverso dal precedente si gioca la mossa corrispondente alla vincita.

Sotto queste ipotesi si osserva che la probabilità di essere nel gioco  $\bar{s}$ , condizionata all'osservazione della mossa corrispondente  $T$  nel turno precedente, è data da:

$$\mathbb{P}(X_i = \bar{s} | O_{i-1} = T) = p\mathbb{P}(X_{i-1} | O_{i-1} = T) + (1-p)\mathbb{P}(X_{i-1} = \underline{s} | O_{i-1} = T) \quad (4.28)$$

Si calcola uno dei delle due probabilità in (4.28):

$$\mathbb{P}(X_n = \bar{s} | O_n = T) = \frac{\mathbb{P}(X_n = \bar{s}, O_n = T)}{\mathbb{P}(O_n = T)} = \frac{\mathbb{P}(X_n = \bar{s}, O_n = T)}{0.5} \quad (4.29)$$

dove  $\mathbb{P}(O_n = T) = \frac{1}{2}$  per simmetria delle probabilità di giocata a seconda delle situazioni. Allo stesso modo

$$\begin{aligned} \mathbb{P}(X_n = \bar{s}, O_n = T) &= \mathbb{P}(X_{n-1} = \bar{s}) \mathbb{P}(X_n = \bar{s}, O_n = T | X_{n-1} = \bar{s}) + \\ &+ \mathbb{P}(X_{n-1} = \underline{s}) \mathbb{P}(X_n = \bar{s}, O_n = T | X_{n-1} = \underline{s}) = \\ &= 0.5 p \tau + 0.5 (1-p) \end{aligned}$$

Quindi

$$\mathbb{P}(X_n = \bar{s} | O_n = T) = p \tau + (1-p). \quad (4.30)$$

In generale si potrebbe assegnare una probabilità anche alla giocata della mossa corrispondente al gioco nel caso di cambio della matrice di payoff, ovvero si introduce la regola che

- se la matrice di gioco è la stessa di quella del turno precedente, si gioca la mossa corrispondente con probabilità  $\rho$ .

Ciò comporta che

$$\mathbb{P}(X_n = \bar{s} | O_n = T) = p \tau + (1-p) \rho. \quad (4.31)$$

A questo punto si osserva che, per ottenere una strategia che non permette alcuna inferenza da parte dell' avversario, ovvero per ottenere

$$\mathbb{P}(X_i = \bar{s} | O_{i-1}) = 0.5 \quad (4.32)$$

bisogna imporre che

$$\mathbb{P}(X_n = \bar{s} | O_n = T) = p \tau + (1 - p) \rho = 0.5 \quad (4.33)$$

Si osserva facilmente, che se volessimo imporre la condizione ( $\tau = \rho$ ), allora l' unica soluzione di (4.33) è  $\tau = \rho = 0.5$ , ovvero una strategia totalmente casuale. Come anticipato, tuttavia, si cercano strategie migliori della casuale. In più, supponiamo che per sua natura,  $\rho$  sia maggiore di  $\tau$ . Quindi imponendo la condizione

$$\rho = \tau + \epsilon, \quad \epsilon > 0 \quad (4.34)$$

e sostituendo alla (4.33) si ottiene che

$$p\tau + (1 - p)(\tau + \epsilon) = 0.5 \Leftrightarrow \tau = 0.5 - (1 - p)\epsilon \quad (4.35)$$

quindi  $\tau < 0.5$ . Ciò comporterebbe che durante la permanenza in un determinato gioco, il giocatore informato dovrebbe giocare peggio che totalmente a caso.

Rimane da considerare l' ipotesi in cui allora il giocatore informato non punti ad una strategia totalmente celante, ma che si accontenti di non far inferire troppo all' avversario.

Si testa questo metodo euristico per il caso  $p = 0.9$ , utilizzando per il giocatore 2 i 2 tipi di strategie già adottate in precedenza: la strategia basata sulle catene di Markov nascoste e la strategia ottimale del caso  $p \in [\frac{1}{2}, \frac{2}{3}]$ .

In prima istanza si considera  $\tau = 0.55$ , per i quali si riportano le percentuali di guadagno in 100 partite da 10000 turni ciascuna in Figura 4.10. Si osserva che il guadagno medio nel caso di giocatore 2 giocante la strategia basata sulle catene di Markov nascoste è inferiore al 25%, guadagno che sarebbe stato ottenuto giocando mosse del tutto casuali.

#### 64. Applicazione di Modelli di Markov nascosti ai Giochi Markoviani.

---

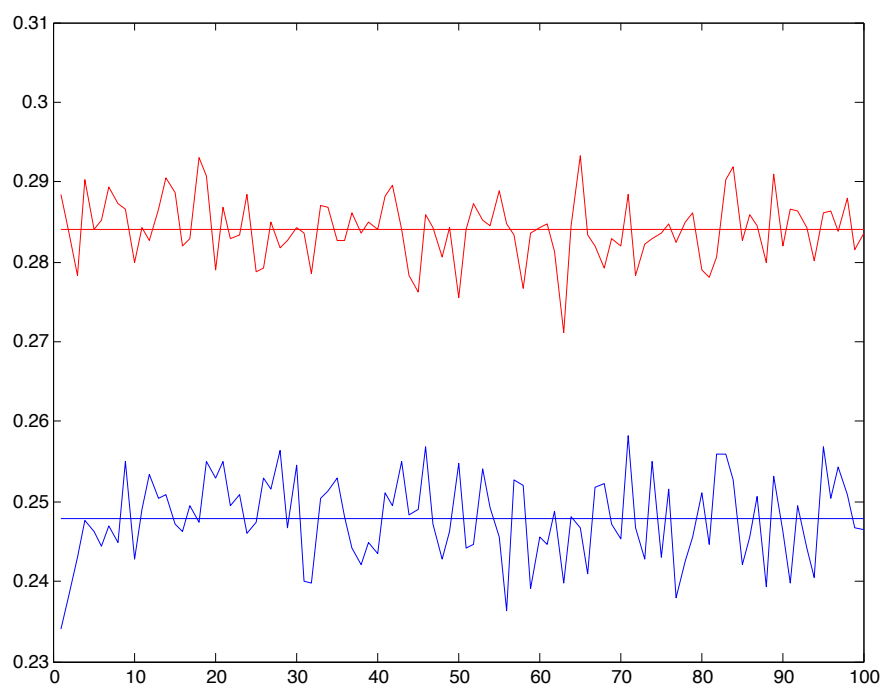


Figura 4.10: Confronto della strategia basata su catene di Markov nascoste (in blu), rispetto alla strategia di Bressaud (in rosso), considerando il giocatore2 con strategia basata su 2 stati consecutivi e con  $\tau = 0.55$



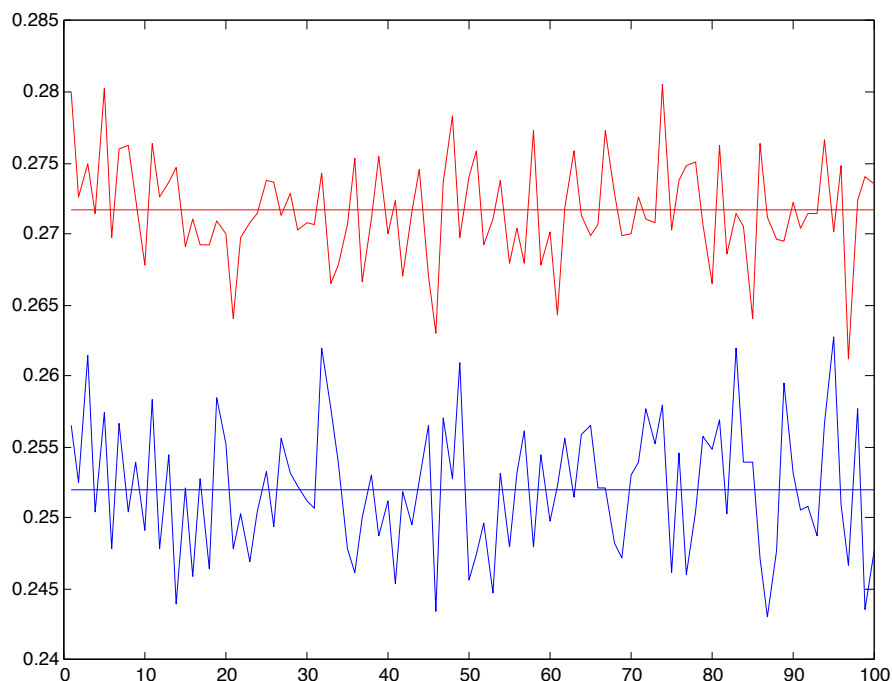


Figura 4.11: Confronto della strategia basata su catene di Markov nascoste (in blu), rispetto alla strategia di Bressaud (in rosso), considerando il giocatore2 con strategia basata su 2 stati consecutivi e con  $\tau = 0.51$

Si prova ad abbassare il valore di  $\tau$ , in modo tale da diminuire la possibile inferenza del giocatore2 durante i periodi di permanenza nella stessa matrice di payoff; ad esempio si testa la strategia con  $\tau = 0.51$ , il cui risultato è riassunto in Figura 4.11. In questo caso si osserva che: la percentuale di guadagno nel caso il giocatore2 giochi una strategia basata su catene di Markov nascosta è di circa il 25,22%, quindi si ha un guadagno rispetto a delle giocate totalmente casuali.

Per  $\tau = 0.505$ , si ha un ulteriore guadagno sulla percentuale di vincita (25,24%), mentre per valori ancora minori testati si osserva una sostanziale stabilizzazione intorno al 25,2%.

Osserviamo che il metodo appena descritto è migliore rispetto alla scelta della mossa solo in base alla matrice di payoff attuale: supponiamo che il giocatore1 giochi secondo uno schema stazionario, ovvero secondo schema di

## 66. Applicazione di Modelli di Markov nascosti ai Giochi Markoviani.

---

Markov nascosto con matrice di emissione

$$B = \begin{bmatrix} \tau & 1 - \tau \\ 1 - \tau & \tau \end{bmatrix} \quad (4.36)$$

e consideriamo che il giocatore2 giochi la strategia basata sulle catene nascoste. Riportiamo in tabella la percentuale di vittoria del giocatore1 al variare di  $\tau$ , per  $p = 0.9$ :

$\tau$	percentuale vittoria (%)
0.51	24.90
0.53	24.97
0.55	25.05
0.6	25.00
0.65	24.98
0.7	24.91

Dalle tabella osserviamo che la percentuale di vincita non si discosta molto dal 25% in ogni caso, mentre precedentemente si è riuscito a ottenere una strategia dal valore di 0,2 punti percentuali in più.

Si precisa che in ogni caso il giocatore2 ha interesse nel non giocare la strategia che è ottimale per  $p \leq \frac{2}{3}$ : per valori così prossimi ad 1, la strategia risulta sempre molto meno efficace della strategia basata sulle catene di Markov nascoste: ad esempio contro un giocatore1 con strategia stazionaria di matrice (4.36) con  $\tau = 0.65$ , la strategia di Bressaud del giocatore2 paga con una percentuale di circa il 30%, quindi ben superiore ad ogni percentuale della tabella sopra riportata.

Si precisa inoltre che le strategie riportate in questa sezione non sono provate con certezza essere le strategie ottimali (tutt' oggi sconosciute) per i due giocatori, tuttavia possiamo concludere che entrambi i giocatori possono considerare di usarle con discreto successo.

# Capitolo 5

## Codici per implementazione in Matlab.

In questa sezione proponiamo delle implementazioni in linguaggio Matlab a riguardo degli algoritmi di base per le catene di Markov nascoste e l' applicazione ai giochi.

Si specificano che nel proseguio si considerano sia gli stati, che lo spazio dei segnali, a valori interi. Per una trattazione più generale sono necessarie funzioni di decodifica che assegnano ad ogni numero, il corrispondente valore in  $S$  spazio degli stati, e  $V$  spazio dei segnali.

### 5.1 Algoritmi di valutazione.

Si implementano le funzioni che calcolano le quantità forward ( $\alpha$ ) e backward ( $\beta$ ), tramite le quali si potranno calcolare la probabilità di un dato vettore di osservazioni.

```
function alpha= prob_avanti(P, b, pi, osserv)
% input: matrici P, b; distribuzione pi ; vettore osserv
% output: matrice di dimensioni "lunghezza(osserv) x N "
% valori di stati e osservazioni sono di tipo "indice"
%          (1:N per gli stati, 1:M per le osservazioni)
```

```

n=length(osserv);
num_stati=length(pi);
i=0;
for i=1:num_stati
    alpha(1,i)= pi(i) * b(i,osserv(1));
end
if (n>1)
    for(tempo = 2: n)
        alpha_temp=alpha;
        stato=0;
        for(stato= 1:num_stati)
            alpha(tempo,stato)= b(stato, osserv(tempo))* ...
                (alpha(tempo-1,:)*P(:, stato));
        end;
    end;
end

```

```

function beta= prob_indietro(P, b, pi, oss)
n=length(oss);
stati=length(pi);
beta(n, :)=ones(1, stati);
for tempo=n-1:-1:1
    for i=1:stati
        q=0;
        for j=1:stati
            q=q+ b(j, oss(tempo+1))* beta(tempo+1,j)* P(i, j);
        end;
        beta(tempo,i)=q;
    end;
end;
end

```

A questo punto il problema del calcolo della probabilità del vettore `osservazioni`, nel modello nascosto  $\lambda = \{\pi, P, b\}$  può essere implementato come segue (algoritmi in avanti e in indietro).

```
function valore_avanti=eval_avanti(P,b,pi,osservazioni)
alpha=prob_avanti(P,b,pi,osservazioni);
valore_avanti=sum(alpha(end,:));
end
```

```
function valore_indietro=eval_indietro(P,b,pi,osservazioni)
beta=prob_indietro(P,b,pi,osservazioni);
valore_indietro=sum(b(:,osservazioni(1)).*beta(1,:).*pi);
end
```

## 5.2 Algoritmo di Viterbi.

Presentiamo una versione basilare dell' algoritmo di Viterbi nell' ambito del problema della decodifica.

```
function sequenza= viterbi(P, b, pi, oss)
% input: matrici P e b
%          vettori pi (distribuzione di prob)
%          oss (osservazioni)
%output: vettore sequenza di best path
stati=length(pi);
lung=length(oss);
delta(1,:)= pi.*b(:,oss(1))';
psi(1,:)=zeros(1, stati);
for t=2:lung
    for j=1:stati
        delta(t, j)=0;
        psi(t,j)=0;
        delta(t,j)= max( delta(t-1, :).* P(:, j)') *b(j, oss(t));
```

```

        [m, psi(t,j)]=max(delta(t-1,:).*P(:,j)');
    end;
end;
sequenza=zeros(1,lung);
[m,sequenza(lung)]=max(delta(lung,:));
for t=lung-1:-1:1
    sequenza(t)=psi(t+1, sequenza(t+1));
end;
end

```

### 5.3 Algoritmi di learning dei parametri.

Presentiamo le funzioni con le formule di ristima di Baum-Welch e l' algoritmo di training basato su queste ultime, nel caso si voglia inferire solo sulla matrice di emissione o su tutti i parametri.

In prima istanza si presenta l' implementazione delle quantità forward e backward scalate. Come osservato nella teoria l' introduzione di algoritmi scalati è fondamentale per la stabilità numerica dell' algoritmo.

```

function [bar_alpha,bar_beta,c]= scaling(P,b,pi, oss)
% input: modello HMM e vettore osservazioni
% output: tutti i valori forward, backward e fattori di scala
[N,M]=size(b);
alpha= zeros (length(oss), length(pi));
bar_alpha=alpha;
for stato=1:N
    alpha(1,stato)= pi(stato)* b(stato, oss(1));
end
c(1)= 1/ sum(alpha(1,:));
bar_alpha(1,:)= c(1)* alpha(1,:);
for t=2:length(oss)
    for stato=1:N

```

```

    alpha(t,stato)= 0;
    for j=1:N
        alpha(t,stato)=alpha(t,stato)+bar_alpha(t-1,j)*...
                                P(j,stato)*b(stato,oss(t));
    end
end
c(t)=1/sum(alpha(t,:));
bar_alpha(t,:)=alpha(t,:)*c(t);
end
    bar_beta=zeros(size(bar_alpha));
beta1=bar_beta;
beta1(end, :)=ones(1,N);
bar_beta(end,:)=beta1(end,:)*c(end);
for t= length(oss)-1:-1:1
    for i=1:N
        beta1(t,i)=0;
        for j=1:N
            beta1(t,i)=beta1(t,i)+P(i,j)*b(j, oss(t+1))* bar_beta(t+1,j);
        end
    end
    bar_beta(t,:)= c(t)*beta1(t,:);
end
end
end

```

A questo punto si riporta la funzione per la ristima di tutti i parametri tramite Baum-Welch.

```

function [P1,b1,pi1]=Baum_Welch_scalato(P,b,pi,oss)
%output: nuova stima parametri HMM
[alpha, beta, c]=alpha_scaling(P,b,pi,oss);
[stati,segnali]=size(b);
for i=1:stati

```

```

    for j=1:stati
        q=0;
        den=0;
        for t=1:length(alpha)-1
            q=q+alpha(t, i)*P(i,j)*b(j, oss(t+1))* beta(t+1,j);
            den=den+ alpha(t, i)* beta(t,i)/c(t);
        end
        P1(i,j)=q/den;
    end
end
for j=1:stati
    for k=1:segnali
        num=0;
        denom=0;
        for t=1:length(oss)
            if oss(t)==k
                num=num+ alpha(t,j)*beta(t,j)/c(t);
            end;
            num=num;
            denom=denom+alpha(t,j)*beta(t,j)/c(t);
        end;
        b1(j,k)=num/denom;
    end;
end;
pi1= alpha(1,:).*beta(1,:);
pi1= pi1/sum(pi1);
end

```

Per quanto riguarda il problema delle strategie del gioco Markoviano, si è utilizzato con successo il training solo sulla matrice di emissione. Riportiamo quindi brevemente due codici per la ristima della sola matrice  $b$  e di una



funzione che applica questa ristima finchè non si raggiungano delle condizioni di uscita:

- superamento del numero massimo di ristime (parametro scelto dall'utente);
- raggiungimento di un appropriata stima della verosimiglianza del modello: il ciclo si interrompe quando la ristima del parametro non aumenta in modo consistente la probabilità del vettore delle osservazioni.

```
function beta_star= Baum_Welch_emiss(P,b,pi,oss, alpha, beta)
%input: parametri HMM, osservazioni, quantità forward e backward
% output: nuova matrice di emissione
stati=length(pi);
segnali=length(b(1,:));
lung=length(oss);
for j=1:stati
    for k=1:segnali
        num=0;
        denom=0;
        for t=1:lung
            if oss(t)==k
                num=num+ alpha(t,j)*beta(t,j);
            end;
            denom=denom+alpha(t,j)*beta(t,j);
        end;
        beta_star(j,k)=num/denom;
    end;
end;
end
end
```

Si applica la funzione appena riportata nel modo seguente, per implementare l' algoritmo scalato:

```
function b=Baum_training_emiss(P,b,pi,oss,contmax,tol)
diffLoglike=1;
cont=0;
while(diffLoglike>=tol && cont<=contmax)
    [alpha_scal, beta_scal, c]=alpha_scaling(P,b,pi,oss);
    Loglike=-sum(log(c));
    for t=1:length(mosse1)
        beta_baum(t,:)=beta_scal(t,+)/c(t);
    end
    b_ciclo=Baum_Welch_emiss(P,b,pi,oss,alpha_scal,beta_baum );
    [alpha_scal, beta_scal, c]=alpha_scaling(P,b,pi,oss);
    newLoglike=-sum(log(c));
    if newLoglike>=Loglike
        diffLoglike=abs(Loglike-newLoglike);
        Loglike=newLoglike;
        b=b_ciclo;
    else
        diffLoglike=eps;
    end
    cont=cont+1;
end
end
```

## Capitolo 6

### Conclusioni e lavori futuri.

In questa trattazione si sono presentati i modelli di Markov nascosti, ed una possibile applicazione al campo della Teoria dei Giochi. Questo approccio si è dimostrato utile per determinare una strategia di gioco euristica nel caso del gioco stocastico introdotto da Renault in [11], nel quale uno dei due giocatori non è al corrente della matrice di gioco effettiva in ogni turno della partita.

Come osservato dalle simulazioni numeriche delle giocate, riassunte nei grafici del Capitolo 4, nei casi in cui esistono strategie d'equilibrio note si ha che l'uso di una strategia basata sul concetto di catena di Markov nascosta da parte del giocatore non informato si rivela essere competitiva con la strategia ottimale, nel caso in cui il giocatore informato decida di giocare la rispettiva strategia d'equilibrio.

Per di più, fermo restando che al giocatore informato convenga giocare la strategia d'equilibrio, nei casi in cui gioca strategie stazionarie, ovvero sceglie le proprie mosse solamente in reazione al gioco corrente, si è osservato che la strategia da noi trovata per il giocatore non informato risponde molto meglio della strategia ottimale.

Per i casi nei quali le strategie d'equilibrio, se esistono, sono sconosciute, abbiamo osservato nella sezione 4.5 che le strategie basate sulle catene di Markov nascoste possono essere applicate da entrambi i giocatori con discre-

to successo.

In conclusione possiamo assumere che un giocatore non informato, che in generale non conosce la strategia dell' avversario informato, può decidere di adottare con successo le strategie di Markov nascoste.

Possibili futuri lavori correlati possono riguardare l' utilizzo delle strategie presentate in questa trattazione applicate a giochi stocastici più generali, ad esempio modelli in cui le singole matrici di payoff della catena di Markov non abbiano strategie dominati, bensì delle strategie miste di equilibrio, e a casi in cui il giocatore non informato non conosce con certezza le probabilità di transizione del processo stocastico dei giochi.

In questo caso, similmente al lavoro di Benevides [14], si potranno costruire dei modelli di Markov nascosti nei quali la matrice di emissione  $b$  è composta dalle probabilità definenti le strategie di equilibrio di ogni singolo gioco, e gli algoritmi di learning dovranno essere applicati su tutti i parametri del modello, non solo sulla matrice di emissione.

# Bibliografia

- [1] S.M.Ross, “ *Introduction to probability Models, 10th Edition.*”
- [2] P.Dymarsky, “ *Hidden Markov Models, Theory and Application.*”
- [3] L.R.Rabiner, B.H.Juang, “ *An Introduction to Hidden Markov Model.* ”
- [4] L.R. Rabiner , “ *A tutorial on Hidden Markov Models and Selected Applications in speech recognition.*”
- [5] E.Levinson, L.R.Rabiner, M.M.Sondhi, “ *An introduction to the Application of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recognition.*”
- [6] L.E.Baum, T.Petrie, ‘ *Statistical Inference for Probabilistic Functions of finite State Markov Chain.*”
- [7] L.E.Baum, T.Petrie, G.Soules, N.Weiss, “ *A maximization technique occurring in the statistical analysis of probabilistic function of Markov Chains.*”
- [8] D.A. Coley, “ *An introduction to Genetic Algorithms for Scientists and Engineers* ”
- [9] J. Xiao, L. Zou, C. Li, “ *Optimization of Hidden Markov Model by a Genetic Algorithm for web Information Extraction*”
- [10] C. Chau, S.Kwong, “ *Optimization of HMM by Genetic Algorithm*”

- [11] J.Renault, “*The value of Markov Chain Game with Lack of Information on One Side*”
- [12] X.Bressaud, A.Quass, “*Asymmetric Warfare*”
- [13] J.Horner, D.Rosenberg e al., “*On a Markov Game with One-Sided Incomplete Information* ”
- [14] M.Benevides, L.Lima e al., “*Using HMM in Strateg Game*”
- [15] V.Frangelli, “*Teoria dei Giochi*”, dispense disponibili on line.
- [16] E.Dekel, D.Fundenberg, D.Levine, “*Learning to Play Bayesian Game*”

# Ringraziamenti.

In primo luogo vorrei ringraziare i docenti dell' università di Bologna dai quali in questi anni ho cercato di apprendere il più possibile, in special modo il professor Massimo Campanino, per la disponibilità e il supporto nel lavoro di questa tesi.

Arrivato al termine di questo percorso di studio vorrei ringraziare tutte le persone che mi sono state vicine, e che hanno contribuito alla mia formazione, quindi in primo luogo i miei genitori, Vincenzo e Maria Pia, che hanno dato sempre tutto pur di non farmi mancare niente sia da un punto di vista materiale che emotivo; da loro ho imparato lo spirito di sacrificio e il valore del lavoro e dell' impegno e spero che un giorno riesca ad essere un modello altrettanto valido per i miei figli.

Oltre ai miei genitori, ringrazio tutta la mia famiglia: i miei fratelli, Daniele e Alberto, e i nonni Fioravante, Lucia, Antonietta e Umberto che sono stati come dei secondi genitori.

Un ringraziamento speciale va a Ludovica che mi ha fatto crescere molto in poco tempo, e che è la persona che ho avuto più vicino in questi anni.

In ultimo vorrei citare tutti gli amici, e i ragazzi grazie ai quali ho trascorso praticamente solo bei momenti: i miei amici di Atri, con i quali sono cresciuto, i coinquilini "bolognesi" che più che semplici conviventi sono stati dei veri e propri amici, e tutte le persone che ho frequentato in questi anni.