

# APPLICAZIONI ALGORITMICHE DELLE CATENE DI MARKOV

Seconda parte delle dispense del corso  
“Modelli Stocastici Markoviani”

Corsi di laurea magistrale di Informatica  
e di Tecnologie dell’Informazione e della Comunicazione

Dipartimento di Scienze dell’Informazione  
Università degli Studi di Milano

Maggio 2008  
Versione preliminare

Massimiliano Goldwurm



# Indice

<b>1</b>	<b>Catene reversibili</b>	<b>5</b>
1.1	Catene reversibili . . . . .	5
1.2	Passeggiate a caso su grafi . . . . .	7
1.2.1	Passeggiate in un grafo completo . . . . .	8
1.2.2	Passeggiate in un cammino semplice . . . . .	9
<b>2</b>	<b>Simulazione di catene di Markov</b>	<b>13</b>
2.1	Generazione casuale non uniforme . . . . .	13
2.2	Algoritmo di simulazione . . . . .	14
2.3	Catene a grado limitato . . . . .	15
<b>3</b>	<b>Generazione casuale mediante catene di Markov</b>	<b>17</b>
3.1	Generazione di insiemi indipendenti . . . . .	17
3.2	Campionatori di Gibbs . . . . .	20
3.3	Generazione di colorazioni di grafi . . . . .	21
3.4	L'algoritmo di Metropolis . . . . .	23
<b>4</b>	<b>Analisi della velocità di convergenza</b>	<b>25</b>
4.1	Approssimazione generale alla distribuzione stazionaria . . . . .	26
4.2	Velocità di convergenza della colorazione di grafi . . . . .	27
<b>5</b>	<b>Approssimazione per problemi di conteggio</b>	<b>31</b>
5.1	Conteggio approssimato di colorazioni . . . . .	33
	<b>Bibliografia</b>	<b>39</b>



# Capitolo 1

## Catene reversibili

In questo capitolo introduciamo anzitutto la nozione di reversibilità di una catena di Markov. Si tratta di una proprietà delle catene che consente in molte applicazioni di determinare in maniera semplice una distribuzione stazionaria e quindi, nei casi ergodici, la distribuzione limite della catena. Nella seconda sezione invece mostreremo alcuni esempi tipici di catene reversibili, legate alle passeggiate a caso su grafi non orientati, che hanno numerosi risvolti applicativi. Il famoso algoritmo di Metropolis, che introdurremo nella sezione 3.4, può essere visto come una variante di tali procedimenti. Un altro esempio di rilievo è dato dalle passeggiate a caso su grafi formati da una semplice lista di vertici; queste infatti consentono di studiare un interessante esempio di algoritmo probabilistico per la soluzione di un problema classico (2-CNF SODD) e di valutarne la probabilità di errore.

### 1.1 Catene reversibili

Una catena di Markov  $\{X_n\}_n$ , definita sullo spazio degli stati  $S = \{1, 2, \dots, k\}$  con matrice di transizione  $P = [p(i, j)]_{i, j \in S}$ , si dice *reversibile* se esiste un vettore stocastico  $\pi$  su  $S$  tale che per ogni  $i, j \in S$

$$\pi_i p(i, j) = \pi_j p(j, i) \quad (1.1)$$

Diciamo anche che  $\pi$  è una distribuzione reversibile per la catena  $\{X_n\}_n$ .

La proprietà appena definita può essere considerata una forma di simmetria della catena, essa equivale infatti a porre

$$\Pr_\pi(X_0 = i, X_1 = j) = \Pr_\pi(X_0 = j, X_1 = i)$$

Più in generale la reversibilità implica che per ogni sequenza finita di stati  $i_0, i_1, \dots, i_k$ ,

$$\Pr_\pi(X_0 = i_0, X_1 = i_1, \dots, X_k = i_k) = \Pr_\pi(X_0 = i_k, X_1 = i_{k-1}, \dots, X_k = i_0)$$

La proprietà fondamentale delle catene reversibili è la seguente.

**Proposizione 1.1** *Se  $\pi$  è una distribuzione reversibile per una catena  $\{X_n\}_n$  allora  $\pi$  è anche distribuzione stazionaria per  $\{X_n\}_n$ .*

*Dimostrazione.* Siano  $S$  e  $P = [p(i, j)]_{i, j \in S}$ , rispettivamente, l'insieme degli stati e la matrice di transizione della catena. Poiché  $\pi$  è una distribuzione reversibile, per ogni  $j \in S$  abbiamo

$$(\pi'P)_j = \sum_{i=1}^m \pi_i p(i, j) = \sum_{i=1}^m \pi_j p(j, i) = \pi_j$$

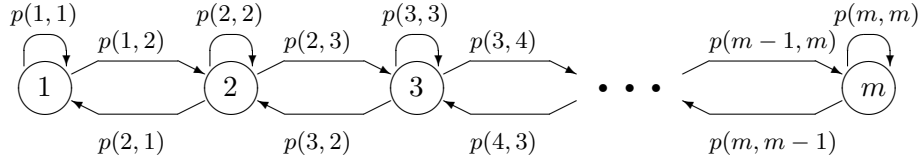
Di conseguenza  $\pi'P = \pi'$  e quindi  $\pi$  risulta stazionaria.  $\square$

Allora, se  $\{X_n\}_n$  è una catena di Markov irriducibile e reversibile, la sua distribuzione stazionaria coincide con la distribuzione reversibile  $\pi$  (che soddisfa le equazioni (1.1)).

Esempi classici di catene reversibili sono forniti dai cosiddetti processi di nascita e morte. Formalmente, un *processo di nascita e morte* è una catena di Markov su un insieme di stati  $S = \{1, 2, \dots, m\}$  con matrice di transizione  $P = [p(i, j)]_{i, j \in S}$ , tale che

$$\begin{aligned} p(i, j) &> 0 \text{ se } |i - j| = 1, \text{ e} \\ p(i, j) &= 0 \text{ se } |i - j| \geq 2. \end{aligned}$$

Il grafo di connessione della catena è descritto dalla seguente figura.



Chiaramente tale catena è irriducibile e risulta inoltre aperiodica se  $p(i, i) > 0$  per qualche  $i \in S$ . Mostriamo ora la sua reversibilità. A tale scopo definiamo  $m$  coefficienti reali positivi  $r_1, r_2, \dots, r_m$ , dove  $r_1 > 0$  è un valore qualsiasi, mentre i successivi sono dati da

$$r_2 = r_1 \frac{p(1, 2)}{p(2, 1)}, \quad r_3 = r_2 \frac{p(2, 3)}{p(3, 2)}, \quad \dots, \quad r_m = r_{m-1} \frac{p(m-1, m)}{p(m, m-1)}.$$

Questo equivale a porre, per ogni  $i = 2, \dots, m$ ,

$$r_i = r_1 \frac{p(1, 2)p(2, 3) \cdots p(i-1, i)}{p(i, i-1) \cdots p(3, 2)p(2, 1)}.$$

Consideriamo ora il vettore stocastico  $\pi$  dato dalle equazioni

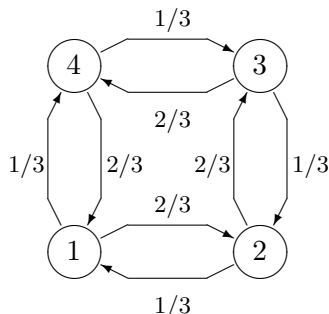
$$\pi = (\pi_1, \pi_2, \dots, \pi_m) = \left( \frac{r_1}{\sum_{i=1}^m r_i}, \frac{r_2}{\sum_{i=1}^m r_i}, \dots, \frac{r_m}{\sum_{i=1}^m r_i} \right)$$

Per la stessa definizione dei coefficienti  $r_i$  è facile verificare che  $\pi$  è una distribuzione reversibile per la catena considerata. Di conseguenza,  $\pi$  è anche l'unica distribuzione stazionaria. Nota che i suoi valori non dipendono dalle probabilità  $p(i, i)$ , per  $i \in S$ .

**Proposizione 1.2** *Sia  $\{X_n\}_n$  un catena di Markov con matrice di transizione  $P$  irriducibile e bistocastica. Se  $\{X_n\}_n$  è reversibile allora  $P$  è simmetrica.*

*Dimostrazione.* Infatti, essendo  $P$  irriducibile e bistocastica, la sua unica distribuzione stazionaria è quella uniforme  $\pi = (1/k, 1/k, \dots, 1/k)$ , dove  $k$  è la dimensione di  $P$ . Inoltre, per la reversibilità della catena  $\pi$  è anche l'unica distribuzione reversibile. Di conseguenza,  $\pi_i P_{ij} = \pi_j P_{ji}$  per ogni coppia di indici  $i, j$ , il che implica  $P_{ij} = P_{ji}$ .  $\square$

La proposizione precedente permette di fornire facilmente esempi di catene che non sono reversibili. È sufficiente considerare matrici irriducibili bistocastiche e non simmetriche. Le corrispondenti catene di Markov non possono essere reversibili. Per esempio, la catena definita dal seguente grafo di connessione non è reversibile.



## 1.2 Passeggiate a caso su grafi

Un altro esempio fondamentale di catena reversibile è fornito dalle passeggiate a caso in un grafo non orientato. Tale catena è definita nel modo seguente. Sia  $G = (V, E)$  un grafo non orientato, con  $V = \{1, 2, \dots, k\}$  insieme dei nodi,  $E$  insieme degli archi dove  $\#E = m$  (cioè supponiamo che la cardinalità di  $E$  sia  $m$ ). Per ogni  $i \in V$  denotiamo inoltre con  $d_i$  il grado di  $i$ , ovvero  $\#\{\{i, j\} \in E \mid j \in V\}$ . Chiaramente abbiamo  $\sum_{i \in V} d_i = 2m$ .

Consideriamo ora una catena di Markov  $\{X_n\}_n$  sullo spazio degli stati  $V$ , con matrice di transizione  $P = [p(i, j)]_{i, j \in V}$  tale che, per ogni  $i, j \in V$ ,

$$p(i, j) = \begin{cases} 1/d_i & \text{se } \{i, j\} \in E \\ 0 & \text{altrimenti} \end{cases}$$

Diciamo che  $\{X_n\}_n$  è la catena delle *passeggiate a caso* su  $G$ . Essa rappresenta un processo naturale di visita dei nodi di grafo a partire da un vertice dato; ad ogni passo il nuovo nodo viene scelto in maniera equiprobabile tra tutti i vertici adiacenti al nodo corrente. Definiamo il vettore stocastico  $\pi$  su  $V$  dato da

$$\pi = \left( \frac{d_1}{2m}, \frac{d_2}{2m}, \dots, \frac{d_k}{2m} \right) \quad (1.2)$$

È facile verificare che  $\pi$  è un vettore reversibile per la catena  $\{X_n\}_n$ . Infatti, per ogni coppia di nodi distinti  $i, j \in V$ , se  $\{i, j\} \notin E$  allora  $\pi_i p(i, j) = 0 = \pi_j p(j, i)$ . Viceversa, se  $\{i, j\} \in E$  si verifica immediatamente che  $\pi_i p(i, j) = \frac{1}{2m} = \pi_j p(j, i)$ .

Inoltre, osserviamo che se  $G$  è connesso la catena è irriducibile. Applicando allora la proposizione 1.2 e le proprietà della sezione ?? otteniamo il seguente risultato.

**Proposizione 1.3** *Se  $G$  è un grafo non orientato connesso allora la catena delle passeggiate a caso su  $G$  gode delle seguenti proprietà:*

- a) *la catena è irriducibile e reversibile,*
- b) *la sua periodicità è al più 2,*
- c) *la sua distribuzione stazionaria è data dal vettore  $\pi$  definito in (1.2),*

d) per ogni nodo  $i$  di  $G$  il tempo medio di rientro in  $i$  è dato da

$$E_i(\tau_i) = 2m/d_i$$

dove  $m$  è il numero di lati di  $G$  e  $d_i$  è il grado di  $i$ .

Solitamente sono due i parametri di interesse nelle passeggiate a caso in un grafo: il tempo medio di passaggio da un nodo  $i$  a un nodo  $j$ , ovvero  $E_i(\tau_j)$ , e il tempo medio necessario per visitare tutti i nodi del grafo, detto anche tempo medio di copertura del grafo. Tali quantità dipendono dalla forma del grafo e qui nel seguito le studiamo per due famiglie particolari di grafi: i grafi completi e i cammini semplici, che rappresentano rispettivamente i grafi connessi con il massimo e con il minimo numero di lati.

### 1.2.1 Passeggiate in un grafo completo

Sia  $G$  un grafo completo di  $k \geq 3$  nodi e sia  $u$  un nodo di  $G$ . Denotiamo con  $\{X_n\}_n$  la catena di Markov delle passeggiate a caso su  $G$  assumendo che lo stato iniziale sia  $u$ . Chiaramente tale catena è irriducibile e aperiodica e, per le proprietà precedenti, la sua distribuzione stazionaria è  $\pi = (1/k, 1/k, \dots, 1/k)$ . Di conseguenza, il tempo medio di rientro in ciascun nodo  $v$  è proprio  $E_v(\tau_v) = k$ , dove con  $\tau_v$  rappresentiamo il tempo di prima entrata in  $v$ .

Vogliamo ora valutare  $E_u(\tau_v)$ , ovvero il tempo medio di prima entrata in un nodo generico  $v$  partendo dal nodo iniziale  $u$  e supponendo che  $v \neq u$ . In questo caso la passeggiata casuale entra in  $v$  al primo passo con probabilità  $\frac{1}{k-1}$  mentre entra in un nodo diverso da  $v$  con probabilità  $\frac{k-2}{k-1}$ ; in quest'ultimo caso il processo si ripete per simmetria con le stesse probabilità. La variabile aleatoria  $\tau_v$  è quindi una geometrica di parametro  $\frac{1}{k-1}$ . Come sappiamo la sua media è  $k-1$  e questo prova la seguente proprietà.

**Proposizione 1.4** Per ogni nodo  $v$  diverso dal nodo iniziale  $u$  abbiamo  $E_u(\tau_v) = k-1$ .

Denotiamo ora con  $\Gamma_G$  il tempo di copertura di  $G$ , ovvero il minimo numero di passi compiuti da  $\{X_n\}_n$  per entrare almeno una volta in ogni stato della catena. Formalmente, si tratta della variabile aleatoria definita da

$$\Gamma_G = \min\{n \in \mathbb{N} \mid \forall v \in V \exists i \leq n : X_i = v\}$$

dove  $V$  rappresenta l'insieme dei nodi di  $G$ . Tale variabile è ben definita perché la famiglia degli stati della catena è composta da un'unica classe essenziale. (quindi con probabilità 1 la catena entrerà in ogni stato).

**Proposizione 1.5** Il tempo medio di copertura di un grafo completo  $G$  di  $k$  nodi è dato da

$$E_u(\Gamma_G) = (k-1) \sum_{i=1}^{k-1} \frac{1}{i} \sim k \log k$$

*Dimostrazione.* È facile verificare che  $\Gamma_G$  è una somma di  $k-1$  variabili aleatorie

$$\Gamma_G = \sum_{i=1}^{k-1} \delta_i$$

dove ogni  $\delta_i$  è il minimo numero di passi compiuti dalla passeggiata a caso, dopo aver visitato esattamente  $i$  nodi distinti, per entrare in un nuovo nodo diverso dai precedenti. Chiaramente  $\delta_1 = 1$  mentre, per ogni  $i = 2, 3, \dots, k-1$ ,  $\delta_i$  è una geometrica di parametro  $\frac{k-i}{k-1}$ . Il valor medio di  $\Gamma_G$  è quindi la somma delle medie

$$\sum_{i=1}^{k-1} \frac{k-1}{k-i} = (k-1) \sum_{i=1}^{k-1} \frac{1}{i}$$

□

Intuitivamente, questo significa che nella passeggiata a caso di un grafo completo, ogni nodo viene visitato in media circa  $\log k$  volte prima che tutti i vertici siano stati visitati.

Osserviamo che le variabili aleatorie  $\delta_i$  sono indipendenti e quindi la varianza di  $\Gamma_G$  può essere ricavata dalla somma delle varianze, ottenendo

$$\text{Var}(\Gamma_G) = (k-1)^2 \sum_{i=1}^{k-1} \frac{1}{i^2} - (k-1) \sum_{i=1}^{k-1} \frac{1}{i} \sim \frac{\pi^2}{6} k^2$$

Vale la pena di osservare infine che, pur essendo una somma di variabili aleatorie indipendenti,  $\Gamma_G$  non converge in distribuzione ad una normale. È stato infatti provato che per ogni  $t \in \mathbb{R}$  vale la relazione

$$\lim_{k \rightarrow +\infty} \frac{\Gamma_G - k \log k}{k} \leq t \} = e^{-e^{-t}}$$

La distribuzione  $e^{-e^{-t}}$  è chiamata distribuzione di Gumbel ed è nota in letteratura come distribuzione limite di statistiche d'ordine.

### 1.2.2 Passeggiate in un cammino semplice

Consideriamo ora un grafo  $G$  formato semplicemente da un cammino semplice di  $k+1$  nodi numerati da 0 a  $k$ .



Consideriamo la catena di Markov  $\{X_n\}_n$  definita da una passeggiata a caso nel grafo. Per ogni  $i \in \{0, 1, \dots, k\}$ , denotiamo con  $u_i$  la variabile aleatoria che rappresenta il numero di passi necessari per raggiungere il nodo  $k$ , supponendo di partire da  $i$ . Ovvero  $u_i = \min\{n \in \mathbb{N} : X_n = k / X_0 = i\}$ . Per semplicità, denotiamo inoltre con  $U_i$  il valor medio di  $u_i$ ,  $U_i = E(u_i)$ . Condizionando sul primo passo della passeggiata è facile verificare che i valori  $U_i$  soddisfano il sistema di equazioni

$$\begin{aligned} U_0 &= 1 + U_1 \\ U_1 &= 1 + \frac{U_0 + U_2}{2} \\ &\dots \\ U_i &= 1 + \frac{U_{i-1} + U_{i+1}}{2} \\ &\dots \\ U_{k-1} &= 1 + \frac{U_{k-2}}{2} \end{aligned}$$

che può essere risolto mediante una semplice sostituzione ottenendo

$$U_i = \sum_{j=i}^{k-1} (2j+1) = k^2 - i^2 \quad (\forall i = 0, 1, \dots, k)$$

**Proposizione 1.6** *Compiendo una passeggiata a caso in un grafo formato da un cammino semplice di  $k$  nodi il numero medio di passi necessari per raggiungere una estremità del grafo a partire da un vertice qualunque è minore o uguale a  $k^2$ .*

Il risultato appena provato può essere utilizzato nell'analisi di un semplice algoritmo probabilistico per risolvere il problema 2-CNF SODD (soddisfacibilità di formule booleane in forma normale 2 congiunta). Con un approccio simile è possibile definire un analogo algoritmo probabilistico per 3-CNF SODD (si veda per esempio [8, sez. 5.3]). Prima di descrivere il problema, ricordiamo che un *letterale* è una variabile Booleana o una sua negazione, mentre una *clausola* è un "or" di letterali, ovvero una formula Booleana del tipo  $(u_1 \vee u_2 \vee \dots \vee u_m)$  dove ogni  $u_i$  è un letterale.

#### PROBLEMA 2-CNF SODD

Istanza : una formula booleana  $\Phi = \bigwedge_{i=1}^m C_i$  su un insieme di variabili  
 $V = \{v_1, v_2, \dots, v_k\}$ , dove ogni  $C_i$  è una clausola di due letterali.

Domanda : esiste un assegnamento di valori booleani alle variabili in  $V$  che rende  $\Phi$  vera?

La procedura che presentiamo è tipico esempio di algoritmo probabilistico *one-sided error*; esso infatti restituisce il valore esatto nel caso in cui la formula  $\Phi$  non ammetta un assegnamento che la rende vera, mentre se un tale assegnamento esiste la procedura può sbagliare con una certa probabilità. Sarà ovviamente opportuno progettare l'algoritmo in modo tale che la probabilità di errore sia piccola a piacere.

L'idea dell'algoritmo è molto semplice: si ripete un certo numero di volte un ciclo principale di istruzioni nel quale partendo da un assegnamento casuale  $A$  di valori alle variabili in  $V$  si verifica se  $A$  soddisfa  $\Phi$ . In caso affermativo abbiamo effettivamente trovato un assegnamento che rende vera la formula. In caso contrario si modifica il valore di una variabile che compare in una clausola non soddisfatta da  $A$ , ripetendo il procedimento per il nuovo assegnamento. L'algoritmo è formalmente descritto dalla seguente procedura che dipende da un parametro di iterazione  $h$ :

Ripeti  $h$  volte il seguente blocco

```

begin
   $i := 1$ 
  scegli un assegnamento  $A$  per  $V$  a caso
  while  $A$  non soddisfa  $\Phi$  e  $i \leq 2k^2$  do
    {
      determina una clausola  $C_j$  non soddisfatta da  $A$ 
      scegli a caso una variabile  $v$  che compare in  $C_j$ 
      cambia l'assegnamento  $A$  attribuendo a  $v$  il valore complementare
       $i := i + 1$ 
    }
  if  $A$  soddisfa  $\Phi$  then { return 1
                          stop
  end
return 0

```

Come vedremo più precisamente nella sezione 2.1, la generazione casuale di un assegnamento  $A$  può essere eseguita in  $O(k)$  passi. Inoltre, verificare se  $A$  soddisfa  $\Phi$  richiede un tempo  $\Theta(m)$ . Il tempo complessivo è quindi dell'ordine  $\Theta(hmk^2)$ .

Studiamo ora la probabilità di errore dell'algoritmo. Se  $\Phi$  non è soddisfatta da alcun assegnamento, l'algoritmo non commette errore perché la procedura termina sempre restituendo la risposta negativa.

Supponiamo ora che  $\Phi$  ammetta un assegnamento che la rende vera. In questo caso è possibile che l'algoritmo non trovi un assegnamento del genere e quindi fornisca una risposta sbagliata. Valutiamo la probabilità  $\eta$  di un simile errore. Il valore  $\eta$  è massimo quando vi è un solo assegnamento  $\alpha$  che rende vera  $\Phi$ . In questo caso possiamo considerare la distanza tra  $\alpha$  e l'assegnamento corrente  $A$  calcolato dalla procedura, cioè il numero di variabili cui i due assegnamenti attribuiscono valore diverso. Chiaramente, tale distanza è un intero compreso tra 0 e  $k$ . Ad ogni iterazione interna essa viene incrementata o diminuita di 1 cambiando il valore di una variabile in una clausola  $C_j$  non soddisfatta da  $A$ . Poiché  $A$  e  $\alpha$  attribuiscono valore diverso ad almeno una delle due variabili di  $C_j$ , la probabilità di diminuire la distanza è almeno  $1/2$  mentre la probabilità di aumentarla è al più  $1/2$ . Possiamo rappresentare l'esecuzione di un ciclo esterno dell'algoritmo come una passeggiata a caso in un cammino semplice formato dalla sequenza di nodi  $\{0, 1, 2, \dots, k\}$ , nella quale si parte da un nodo qualsiasi  $i$  e ci si arresta non appena si giunge nel nodo 0.

Ne segue che la probabilità di errore in una singola iterazione del ciclo più esterno è minore o uguale alla probabilità che in una passeggiata a caso in una sequenza di  $k + 1$  nodi non si raggiunga il nodo 0 entro i primi  $2k^2$  passi. Poiché l'algoritmo esegue  $h$  cicli esterni, denotando con  $\{X_n\}_n$  la catena di Markov che rappresenta tale passeggiata, abbiamo

$$\eta \leq (\Pr\{X_n \neq 0, \forall n \leq 2k^2\})^h$$

Qualunque sia il nodo di partenza  $i$  (ottenuto dalla generazione di un assegnamento a caso), denotiamo con  $u_i$  la variabile aleatoria che rappresenta il numero di passi necessari per raggiungere 0. Per la proposizione 1.6 sappiamo che  $E(u_i) \leq k^2$  per qualunque  $i$ . Quindi per la disuguaglianza di Markov abbiamo

$$\Pr\{X_n \neq 0, \forall n \leq 2k^2\} = \Pr(u_i > 2k^2) \leq \frac{E(u_i)}{2k^2} \leq \frac{1}{2}$$

Dalle due disuguaglianze precedenti si ricava

$$\eta \leq (1/2)^h$$

È quindi sufficiente compiere  $h \geq \log_2 \frac{1}{\varepsilon}$  iterazioni principali nella procedura per ottenere una probabilità di errore  $\eta \leq \varepsilon$ , per qualunque  $\varepsilon > 0$ .



## Capitolo 2

# Simulazione di catene di Markov

In questo capitolo vogliamo descrivere l'algoritmo tradizionale di simulazione di una catena di Markov, supponendo sempre che la catena sia finita e omogenea. Come è facile immaginare l'algoritmo utilizza ripetutamente un procedimento di generazione casuale di elementi estratti da un insieme finito secondo un'arbitraria distribuzione di probabilità. Per questo motivo descriviamo innanzitutto il metodo classico utilizzato generalmente per risolvere questo problema. Tale metodo è ampiamente utilizzato come sottoprogramma fondamentale nella progettazione di algoritmi per la generazione casuale delle più comuni strutture combinatorie [6].

### 2.1 Generazione casuale non uniforme

Dato un intero positivo  $k$ , consideriamo l'insieme  $S = \{1, 2, \dots, k\}$  e una qualsiasi distribuzione di probabilità su  $S$ , ovvero una funzione  $p : S \rightarrow [0, 1]$  tale che  $p(1) + p(2) + \dots + p(k) = 1$ . Chiaramente possiamo considerare  $p$  come un vettore stocastico

$$p = (p(1), p(2), \dots, p(k)) \in [0, 1]^k$$

Vogliamo generare un elemento a caso in  $S$  secondo la distribuzione  $p$ . Più precisamente vogliamo definire un algoritmo probabilistico che su input  $p$  restituisca un valore  $y \in S$  tale che, per ogni  $i \in S$ ,

$$\Pr(y = i) = p(i)$$

La procedura che illustriamo è semplice e intuitiva: si genera a caso un numero reale  $x$  uniformemente distribuito nell'intervallo  $[0, 1]$  e quindi si determina il minimo  $i \in S$  tale che

$$x \leq p(1) + p(2) + \dots + p(i) \tag{2.1}$$

Qui supponiamo di poter utilizzare una funzione  $random[0, 1]$  in grado di generare un valore reale nell'intervallo  $[0, 1]$  secondo la distribuzione uniforme. Ricordiamo che gran parte dei linguaggi e dei sistemi di manipolazione simbolica dispongono di funzioni in grado di eseguire efficacemente un simile calcolo.

Il procedimento appena descritto è giustificato dalla seguente considerazione: la distribuzione  $p$  può essere vista come una partizione dell'intervallo  $[0, 1]$  in  $k$  intervallini consecutivi  $t_1, t_2, \dots, t_k$ , di dimensione  $p(1), p(2), \dots, p(k)$ , rispettivamente, come mostrato nella seguente figura.



Quindi, scegliendo un valore reale  $x$  uniformemente distribuito in  $[0, 1]$ , la probabilità che  $x$  cada nell'intervallo  $t_i$  è proprio uguale a  $p(i)$ , per ogni  $i = 1, 2, \dots, k$ . Si tratta quindi di trovare l'intervallo nel quale  $x$  è caduto; il metodo più semplice è quello di determinare il minimo  $i$  che soddisfi la relazione (2.1).

Formalmente l'algoritmo è descritto dalla seguente procedura.

```

Procedure Genera( $p$ )
begin
   $x := \text{random}[0, 1]$ 
   $i := 1$ 
   $u := p(1)$ 
  while  $u < x$  do
    {
       $i := i + 1$ 
       $u := u + p(i)$ 
    }
  return  $i$ 
end

```

Nella valutazione del tempo di calcolo e dello spazio di memoria richiesti assumeremo sempre un criterio di costo uniforme (si veda per esempio [1, 4, 2]). In queste ipotesi è ragionevole assumere che anche il calcolo della funzione  $\text{random}[0, 1]$  richieda tempo costante. Ne segue che il tempo di calcolo richiesto dalla procedura di generazione casuale appena descritta è  $O(k)$ . Inoltre, anche lo spazio di memoria richiesto dall'algoritmo è  $O(k)$ , poiché per l'esecuzione del procedimento dobbiamo mantenere il vettore stocastico  $p$ .

## 2.2 Algoritmo di simulazione

Consideriamo ora una catena di Markov  $\{X_n\}$  definita su un insieme di stati  $S = \{1, 2, \dots, k\}$ , con distribuzione iniziale  $\mu$  e matrice di transizione  $P = [p(i, j)]_{i, j \in S}$ . Vogliamo generare una istanza delle prime  $n$  transizioni compiute dalla catena a partire da uno stato iniziale, per un qualsiasi  $n \in \mathbb{N}$ .

L'algoritmo che descriviamo utilizza direttamente i parametri  $k, \mu$  e  $P$  che definiscono la catena e che supponiamo letti e memorizzati in una fase iniziale di precomputazione. La procedura quindi riceve in input l'intero  $n$  e, dopo aver generato lo stato iniziale secondo la distribuzione  $\mu$ , determina i successivi  $n$  passi scegliendo il nuovo stato corrente mediante le righe della matrice  $P$ . La procedura restituisce gli  $n + 1$  stati consecutivi visitati durante la simulazione.

```

Procedure Simula( $n$ )
begin
   $d := \mu$ 
  for  $i = 0, 1, \dots, n$  do

```

```

      {
        j := Genera(d)
        stampa j
        d := (p(j, 1), p(j, 2), ..., p(j, k))
      }
end

```

La procedura è chiaramente basata sull'algoritmo di generazione non uniforme presentato nella sezione precedente. Per svolgere l'analisi dei tempi di calcolo e dello spazio di memoria richiesti dall'algoritmo, dobbiamo innanzitutto fissare le dimensioni dell'input che nel nostro caso dipende in modo naturale da due parametri: il numero degli stati  $k$  e il numero di passi della simulazione  $n$ . È facile verificare che il tempo di calcolo richiesto dalla procedura `Simula` (sempre secondo un criterio di costo uniforme) è  $O(nk)$ . A tale quantità dobbiamo aggiungere il tempo necessario per leggere e memorizzare i parametri che definiscono la catena di Markov: questo richiede un tempo  $O(k^2)$  se manteniamo in memoria l'intera matrice  $P$ . In questo caso è chiaro che anche lo spazio richiesto dall'algoritmo è  $O(k^2)$ . Altrimenti, se possiamo calcolare di volta in volta il vettore delle probabilità necessario (senza memorizzare l'intera matrice), il procedimento richiede solo uno spazio  $O(k)$ .

## 2.3 Catene a grado limitato

Come abbiamo osservato, la procedura descritta nella sezione precedente richiede (nel caso peggiore) uno spazio di memoria almeno proporzionale al numero di stati. In molte applicazioni tuttavia tale numero è troppo elevato e rende l'algoritmo del tutto inefficiente. Vedremo nel capitolo 3 come, in molte applicazioni rilevanti, il numero degli stati della catena sia esponenziale rispetto alle dimensioni del problema originale. Per questo motivo è importante definire procedure di simulazione che non dipendano direttamente dal numero di stati e, in particolare, che evitino di memorizzare l'intera matrice di transizione.

Possiamo definire una procedura di questo genere quando, per ogni stato, esiste un numero limitato (facilmente calcolabile) di stati raggiungibili in un passo; in questo caso il grafo di connessione della catena ha grado piccolo rispetto al numero di stati. In questa situazione, possiamo limitarci a mantenere, per ogni stato corrente solo le probabilità di transizione negli stati adiacenti, che vengono calcolate di volta in volta. Così il tempo di calcolo e lo spazio di memoria necessari per eseguire la procedura non dipendono più dal numero di stati ma solo dal numero di passi della simulazione e dal grado del grafo di connessione della catena.

Per formalizzare la descrizione dell'algoritmo in questo caso, per ogni stato  $q \in Q$ , denotiamo con  $A_q$  l'insieme  $A_q = \{r \in S \mid p(q, r) > 0\}$ . Chiamiamo inoltre *grado* della catena il massimo delle cardinalità degli insiemi  $A_q$  e lo denotiamo con  $g$ , quindi  $g = \max\{\#A_q : q \in S\}$ .

L'algoritmo può essere allora descritto dalla seguente procedura nella quale si suppone di poter sempre calcolare in un tempo  $O(g)$  l'insieme  $A_q$  e la distribuzione  $p_q = \{p(q, r)\}_{r \in A_q}$ , per ogni  $q \in S$ . Per semplicità supporremo inoltre che lo stato iniziale sia un valore  $q_0 \in S$  fissato a priori.

```

Procedure Simula( $n, q_0$ )
begin
   $q := q_0$ 
  for  $i = 1, 2, \dots, n$  do

```

```

      {
        calcola la distribuzione  $p := \{p(q, r)\}_{r \in A_q}$ 
         $r := \text{Genera}(p)$ 
        stampa  $r$ 
         $q := r$ 
      }
end

```

Nelle ipotesi date è evidente che il tempo di calcolo è  $O/ng$ ). Inoltre, non è più necessario mantenere in memoria la matrice  $P$  e neppure un vettore stocastico di dimensione  $k$  come avveniva nell'algoritmo precedente. Nel nostro caso il vettore stocastico, di volta in volta utilizzato nella generazione casuale, ha una dimensione limitata dal valore di  $g$ . Di conseguenza lo spazio necessario per eseguire la procedura è ridotto a  $O/g$ .

La complessità dell'algoritmo risulta quindi nettamente migliore del precedente proprio quando il valore di  $g$  è molto piccolo rispetto a quello di  $k$ . Questo si verifica per esempio per le catene di Markov definite nella sezione ??, relative alla rovina del giocatore o al moto rettilineo di una particella, oppure per i tradizionali processi di nascita e morte, descritti nella sezione 1.1, o ancora nelle passeggiate casuali considerate nella sezione 1.2.2. In tutti questi casi il grado della catena è costante (al più 3), indipendentemente dal numero di stati: quindi il tempo di calcolo richiesto dalla simulazione diventa  $O/n$  e lo spazio di memoria  $O/1$ .

Altri esempi significativi saranno considerati nel capitolo 3, dove presenteremo catene di Markov che hanno un grado logaritmico rispetto al numero di stati. In questi casi il tempo di calcolo diventa  $O/n \log k$  e lo spazio di memoria  $O/\log k$ .

## Capitolo 3

# Generazione casuale mediante catene di Markov

In questo capitolo illustriamo un metodo naturale per la generazione casuale basata su catene di Markov ergodiche. Supponiamo di voler generare un elemento a caso da un insieme finito  $S$  secondo una distribuzione di probabilità  $\pi$  definita su  $S$ . In alcuni casi l'algoritmo tradizionale illustrato nella sezione 2.1 non è facilmente utilizzabile. Questo capita tipicamente quando l'insieme  $S$  non è ben definito, oppure è difficile calcolare la probabilità  $\pi_i$  dei suoi elementi  $i \in S$ . In questi casi un metodo alternativo spesso utilizzato consiste nel definire una catena di Markov irriducibile e aperiodica  $\{X_n\}$  sull'insieme di stati  $S$  che abbia  $\pi$  come distribuzione stazionaria; poiché la catena è ergodica, sappiamo che qualunque sia il valore di  $X_0$ , per  $n$  grande la probabilità che  $X_n = i$  approssima  $\pi_i$  per ogni stato  $i$ . Possiamo quindi simulare la catena a partire da uno stato qualunque, per un numero di passi  $n$  abbastanza elevato, restituendo lo stato raggiunto al passo  $n$ -esimo. In letteratura si dice che una tale catena è un MCMC corretto per la distribuzione  $\pi$ , dove con MCMC si intende “Markov Chain Monte Carlo” (metodo Monte Carlo basato su catena di Markov).

Per descrivere questo metodo e presentare alcune delle problematiche che si incontrano nella sua applicazione, presentiamo un esempio concreto che può essere considerato tipico di questo approccio.

### 3.1 Generazione di insiemi indipendenti

Ricordiamo innanzitutto che in un grafo non orientato  $G = (V, E)$  un insieme indipendente è un sottoinsieme di vertici  $A \subseteq V$  tra i quali non vi sono lati, ovvero, per ogni  $u, v \in A$  si verifica  $\{u, v\} \notin E$ . Nota che il numero di insiemi indipendenti può essere esponenziale rispetto alle dimensioni del grafo (numero dei nodi). Di conseguenza, se vogliamo generare un insieme indipendente secondo la distribuzione uniforme, l'algoritmo presentato nella sezione 2.1 non è sempre utilizzabile perché può richiedere un tempo di calcolo troppo elevato.

Per inciso, ricordiamo che molti problemi tradizionali definiti su questi insiemi sono di difficile soluzione e rappresentano classici esempi ampiamente studiati nell'area della complessità computazionale. Per esempio, è ben noto che il problema di stabilire se in un grafo non orientato esiste un insieme indipendente di dimensione data è NP-completo. Di conseguenza, calcolare il numero di insiemi indipendenti di data dimensione in un grafo è #P-completo, così come risulta

NP-hard determinare il massimo insieme indipendente in un grafo. Inoltre, si può provare che lo stesso problema di ottimizzazione non è approssimabile in tempo polinomiale con un errore relativo minore di una costante fissata (a meno che  $P=NP$ ). Si tratta quindi di problemi difficili che non ammettono algoritmi di approssimazione efficienti [4].

Definiamo allora una catena di Markov per generare a caso in modo uniforme un insieme indipendente in un grafo non orientato  $G = (V, E)$ . Supponiamo che  $k$  sia il numero dei nodi di  $G$ , in simboli  $k = \#V$ . Denotiamo con  $S$  la famiglia di tutti gli insiemi indipendenti in  $G = (V, E)$ ,

$$S = \{A \subseteq V \mid \forall u, v \in A \{u, v\} \notin E\}$$

Denotiamo inoltre con  $Z_G$  la cardinalità di  $S$  e rappresentiamo con  $\pi$  la distribuzione di probabilità uniforme su  $S$ , ovvero la funzione

$$\pi : S \longrightarrow [0, 1] \text{ tale che, per ogni } A \in S, \pi(A) = \frac{1}{Z_G}$$

La catena di Markov  $\{X_n\}$  è definita quindi assumendo  $S$  come insieme di stati e fissando la probabilità di transizione da uno stato all'altro attraverso la seguente procedura, nella quale  $A$  è lo stato corrente e  $B$  lo stato successivo. In pratica, si sceglie in  $V$  un nodo a caso in maniera uniforme e si lancia una moneta (non distorta): se il risultato del lancio è testa e  $v$  non possiede nodi adiacenti in  $A$ , allora si aggiunge  $v$  ad  $A$ , altrimenti si toglie  $v$  da  $A$ .

**begin**

```

  A := insieme indipendente corrente
  scegli v ∈ V a caso secondo la distribuzione uniforme
  scegli a caso c in {0, 1} in modo equiprobabile
  if c = 1 ∧ (∀w ∈ A {w, v} ∉ E)
    then B := A ∪ {v}
    else B := A \ {v}
  return B

```

**end**

Nota che il nuovo stato  $B$  coincide con lo stato corrente  $A$  ogniqualvolta il risultato del test è positivo e  $v$  appartiene ad  $A$ , oppure il risultato del test è negativo e  $v$  non appartiene ad  $A$ . In ogni caso, la differenza tra uno stato corrente e il successivo consiste al più di un solo nodo.

Più precisamente, denotiamo con  $A \div B$  la differenza simmetrica tra due insiemi  $A$  e  $B$ , ovvero  $A \div B = (A \cup B) \setminus (A \cap B)$ . Allora è facile verificare che, per ogni coppia di stati  $A, B \in S$ , la probabilità  $P(A, B)$  di passare da  $A$  a  $B$  in un passo è data da

$$P(A, B) = \begin{cases} 0 & \text{se } \#(A \div B) > 1 \\ \frac{1}{2k} & \text{se } \#(A \div B) = 1 \\ 1 - \frac{\#\{C \in S \mid \#(A \div C) = 1\}}{2k} & \text{se } A = B \end{cases}$$

**Proposizione 3.1** *La matrice di transizione  $P = [P(A, B)]_{A, B \in S}$  è irriducibile e aperiodica. Inoltre la distribuzione  $\pi$  è reversibile per la catena  $\{X_n\}$  definita sopra.*

*Dimostrazione.* Per dimostrare l'irriducibilità di  $P$ , consideriamo due stati qualsiasi  $A, B \in S$  e siano  $C = A \setminus B$  e  $D = B \setminus A$ . Allora un cammino da  $A$  a  $B$  può essere facilmente

costruito togliendo prima tutti i nodi in  $C$  e poi aggiungendo tutti quelli in  $D$ . Per la definizione della catena è facile verificare che ciascuno di questi passi ha probabilità non nulla e quindi la probabilità dell'intero cammino è positiva. Questo implica che per un opportuno intero  $n > 0$  abbiamo  $P^n(A, B) > 0$  e di conseguenza la matrice è irriducibile.

Osserviamo inoltre che la matrice è chiaramente aperiodica poiché  $P(A, A) > 0$  per ogni stato  $A \in S$ .

Infine per provare la reversibilità di  $\pi$  basta dimostrare che per ogni  $A, B \in S$

$$\pi(A)P(A, B) = \pi(B)P(B, A)$$

Questa equazione segue facilmente dalla definizione della matrice  $P$  data sopra e dal fatto che  $\pi(A) = \pi(B)$ .  $\square$

Questo risultato mostra che la catena  $\{X_n\}$  è ergodica e che  $\pi$  è l'unica distribuzione stazionaria. Di conseguenza, per ogni coppia di insiemi indipendenti  $A, B$  di  $G$  abbiamo

$$\lim_{n \rightarrow +\infty} \Pr_A(X_n = B) = \pi(B)$$

La catena definisce quindi un algoritmo MCMC corretto per generare un insieme indipendente di un grafo secondo una distribuzione uniforme. La procedura è del tutto naturale: si sceglie un insieme indipendente qualsiasi, anche formato da un solo nodo e, a partire da questo si simula la catena per un numero di passi  $n$  grande abbastanza, restituendo in uscita l'insieme indipendente ottenuto. Nel capitolo 4 tratteremo il problema della velocità di convergenza, ovvero il problema di determinare il numero  $n$  di passi sufficienti a garantire una buona approssimazione della probabilità di generazione alla distribuzione stazionaria.

Usando la funzione random, la procedura può essere descritta nel modo seguente:

Generatore( $V, E$ )

**begin**

  assumi  $V = \{v_1, v_2, \dots, v_k\}$

  scegli un nodo  $v_i \in V$  qualsiasi

$A := \{v_i\}$

  scegli  $n \in \mathbb{N}$  abbastanza grande

**for**  $i = 1, 2, \dots, n$  **do**

**begin**

$r := \text{random}(0, 1]$

      determina  $j \in \{1, 2, \dots, k\}$  tale che  $\frac{j-1}{k} < r \leq \frac{j}{k}$

$u := \text{random}(0, 1]$

**if**  $u \leq 1/2 \wedge (v_j \text{ non ha nodi adiacenti in } A)$

**then**  $A := A \cup \{v_j\}$

**else**  $A := A \setminus \{v_j\}$

**end**

**return**  $A$

**end**

### 3.2 Campionatori di Gibbs

L'esempio presentato nella sezione precedente può anche essere considerato come un tipico campionatore di Gibbs (Gibbs sampler). Tali algoritmi sono generatori casuali di funzioni che dipendono da una distribuzione fissata e sono definiti da una opportuna catena di Markov.

Per definire formalmente il problema, consideriamo due insiemi finiti arbitrari  $V$  e  $R$ , entrambi di cardinalità maggiore di 1, ponendo  $k = \#V$  e  $q = \#R$ . Denotiamo con  $R^V$  l'insieme delle funzioni definite su  $V$  a valori in  $R$

$$R^V = \{f : V \rightarrow R\}$$

Sia inoltre  $\pi$  una distribuzione di probabilità su  $R^V$ . Nota che la cardinalità di  $R^V$  è  $q^k$ , un valore esponenziale rispetto a  $k$ . Quindi per generare un elemento a caso in tale insieme secondo una distribuzione fissata, avendo in input  $V$  e  $R$ , le procedure viste nella sezione 2.1 non sono efficienti.

Osserva anche che il problema di generare a caso un insieme indipendente in un grafo  $G = (V, E)$  può essere ricondotto al problema precedente: basta porre  $R = \{0, 1\}$  e denotare con  $\pi$  la distribuzione che assegna ugual probabilità a tutte le  $f \in \{0, 1\}^V$  che sono funzioni caratteristiche di insiemi indipendenti e assegna invece probabilità 0 ad ogni altra funzione in  $\{0, 1\}^V$ .

Un campionatore di Gibbs per generare un elemento di  $R^V$  secondo una distribuzione  $\pi$  è definito da una catena di Markov sull'insieme degli stati

$$S = \{A \in R^V : \pi(A) > 0\}$$

Il nuovo stato della catena viene ottenuto da uno stato corrente  $A \in S$  estraendo a caso in modo uniforme un elemento  $v \in V$  e scegliendo per  $v$  un nuovo valore  $c \in R$  con probabilità  $\pi$  condizionata a conservare uguali ad  $A$  i valori degli altri elementi di  $V$ . Più formalmente, il passo generico della catena da uno stato  $A$  a uno stato  $B$  è descritto dalla seguente procedura:

**begin**

$A :=$  stato corrente

scegli  $v \in V$  a caso secondo la distribuzione uniforme

scegli  $c \in R$  a caso con probabilità'

$$\Pr(c) = \pi(U \in R^V : U(v) = c \mid \forall w \neq v U(w) = A(w))$$

definisci il nuovo stato  $B \in R^V$  nel modo seguente:

$$B(v) = c$$

$$B(w) = A(w) \text{ per tutti i } w \in V \text{ tali che } w \neq v$$

**end**

Nota che il nuovo stato  $B$  differisce da  $A$  al più per il solo valore attribuito all'elemento  $v$  estratto. Inoltre, vi è sempre una probabilità non nulla che  $B$  coincida con  $A$  poiché  $\pi(A) > 0$  e  $A(v)$  è un colore ammissibile per  $v$ .

Per ogni  $A, B \in S$ , denotiamo ora con  $P(A, B)$  la probabilità di scegliere il nuovo stato  $B$  partendo da  $A$ . La matrice  $P = [P(A, B)]_{A, B \in S}$  è quindi la matrice di transizione della catena.

**Proposizione 3.2** *La distribuzione  $\pi$  è reversibile per la catena di Markov definita sopra.*

*Dimostrazione.* Vogliamo provare che per ogni  $A, B \in S$ , vale la relazione

$$\pi(A) \cdot P(A, B) = \pi(B) \cdot P(B, A)$$

Chiaramente se  $A = B$  la proprietà è ovvia. Lo stesso vale se  $A$  e  $B$  si differenziano per più di un valore, infatti in questo caso abbiamo  $P(A, B) = 0 = P(B, A)$ . Se invece  $A$  e  $B$  si differenziano per il valore attribuito a un solo elemento  $v \in V$ , allora per la definizione del passo generico della catena abbiamo

$$P(A, B) = \frac{1}{k} \cdot \frac{\pi(B)}{\pi(U \in R^V : U(w) = A(w) \forall w \neq v)}$$

e

$$P(B, A) = \frac{1}{k} \cdot \frac{\pi(A)}{\pi(U \in R^V : U(w) = B(w) \forall w \neq v)}$$

Poiché in entrambe le espressioni precedenti i denominatori sono uguali, con una semplice sostituzione si verifica immediatamente che  $\pi(A) \cdot P(A, B) = \pi(B) \cdot P(B, A)$ .  $\square$

Di conseguenza, se la catena è irriducibile, la matrice  $P$  risulta primitiva e  $\pi$  è quindi l'unica distribuzione stazionaria.

Il campionatore di Gibbs per  $\pi$  è dunque definito dall'algoritmo che su input  $V$  e  $R$  sceglie anzitutto una funzione iniziale  $I \in R^V$  tale che  $\pi(I) > 0$ . Quindi esegue  $n$  passi della catena di Markov sopra descritta a partire da  $I$ , per un opportuno intero  $n$  abbastanza grande. Se la matrice è irriducibile, siamo sicuri che la distribuzione di probabilità dello stato raggiunto al passo  $n$ -esimo approssimi  $\pi$ . Si lascia per esercizio il compito di dimostrare che il generatore di insiemi indipendenti descritto nella sezione 3.1 è un campionatore di Gibbs.

Una variante del metodo sopra illustrato prevede di selezionare ciclicamente l'elemento  $v$  in  $V$  in modo tale che in ogni sequenza di  $k$  passi consecutivi tutti gli elementi siano stati estratti esattamente una volta. La selezione diventa quindi deterministica e non più casuale. Più precisamente, posto  $V = \{v_1, v_2, \dots, v_k\}$ , al passo  $n$ -esimo la catena sceglie un nuovo valore per l'elemento  $v_j$  di indice  $j = 1 + [n - 1]_k$ , dove con  $[n - 1]_k$  denotiamo il resto della divisione di  $n - 1$  per  $k$ . In questo caso il campionatore di Gibbs è detto *ciclico*. Il passo casuale è qui ristretto alla scelta del valore del nodo estratto, scelta che viene compiuta nello stesso modo precedente. La catena di Markov che si ottiene in questo caso non è più omogenea. Essa è definita da  $k$  matrici stocastiche  $\{P_i\}_{i=1, \dots, k}$ , una per ogni elemento  $v_i$ . È facile tuttavia verificare che  $\pi$  è distribuzione reversibile per ciascuna matrice  $P_i$  e quindi rimane distribuzione stazionaria della catena.

### 3.3 Generazione di colorazioni di grafi

Un altro classico esempio di campionatore Gibbs è dato da un noto algoritmo per la generazione casuale di una colorazione di un grafo. In questa sezione descriviamo tale procedura, apparsa in letteratura in [11, 13], che prevede anche una variante ciclica.

Il problema è definito da un grafo non orientato  $G = (V, E)$  di  $k$  nodi e da un insieme di colori  $Q = \{1, 2, \dots, q\}$ . Come è noto, una  $q$ -colorazione di  $G$  è una funzione  $f : V \rightarrow Q$  tale che, per ogni  $\{u, v\} \in E$ ,  $f(u) \neq f(v)$ . Denotiamo con  $Z_{G, q}$  il numero di  $q$ -colorazioni di  $G$ .

Inoltre, con  $\pi$  rappresentiamo la distribuzione uniforme sulle  $q$ -colorazioni, ovvero la funzione  $\pi : Q^V \rightarrow [0, 1]$  tale che, per ogni  $f \in Q^V$ ,

$$\pi(f) = \begin{cases} \frac{1}{Z_{G,q}} & \text{se } f \text{ e' una } q\text{-colorazione di } G \\ 0 & \text{altrimenti} \end{cases}$$

Il nostro obiettivo è quello di generare una funzione  $f \in Q^V$  secondo la distribuzione di probabilità  $\pi$ .

È bene ricordare che anche in questo caso ci troviamo di fronte a un problema considerato generalmente difficile. Innanzitutto non è detto che un grafo non orientato  $G$  ammetta una  $q$ -colorazione. Anzi, lo stesso problema di stabilire se un grafo ammette una  $q$ -colorazione è NP-completo. Di conseguenza risulta #P-completo il problema di calcolare  $Z_{G,q}$  su input  $G$  e  $q$ ; così come è NP-hard il problema di determinare il minimo numero di colori necessari per colorare un grafo. Anche in questo caso il problema di ottimizzazione non è risolubile in tempo polinomiale in modo approssimato con errore relativo limitato (a meno che  $P = NP$ ). Si tratta quindi di problemi difficili per i quali non sono noti nemmeno algoritmi polinomiali di approssimazione. Infine il problema di decisione resta difficile anche in casi particolari. Per esempio, è NP-completo stabilire se un grafo planare è 3-colorabile (mentre è noto che ogni grafo planare ammette una 4-colorazione).

L'idea dell'algoritmo per il nostro problema è quella di definire una catena di Markov sull'insieme degli stati formato dalle  $q$ -colorazioni di  $G$  e quindi, come al solito, simulare la catena per un numero opportuno di passi. Osserviamo subito che la scelta dello stato iniziale può non essere banale se il numero di colori è piccolo. Addirittura, potrebbero non esistere  $q$  colorazioni per il grafo dato oppure, anche quando queste esistono, se il numero dei colori non è abbastanza grande può darsi sia computazionalmente difficile calcolarne una. Per questo supponiamo che si possa sempre facilmente determinare una  $q$ -colorazione di partenza. È facile verificare che questo avviene sicuramente se  $q$  è maggiore del grado di ogni nodo del grafo.

Quindi, una volta fissata la  $q$ -colorazione iniziale la catena di Markov passa da uno stato corrente  $f$  allo stato successivo  $g$  mediante il seguente procedimento:

**begin**

sia  $f : V \rightarrow Q$  la  $q$ -colorazione di  $G$  corrente  
 scegli a caso  $v \in V$  secondo la distribuzione uniforme  
 calcola l'insieme  $U = \{c \in Q : f(w) \neq c \text{ per ogni } w \text{ vicino di } v\}$   
 (nota che  $U \neq \emptyset$  perche'  $f(v) \in U$ )  
 scegli a caso  $c \in U$  secondo la distribuzione uniforme  
 definisci la nuova  $q$ -colorazione  $g$  nel modo seguente:

$$\begin{aligned} g(v) &= c \\ g(w) &= f(w) \text{ per tutti i } w \in V \text{ diversi da } v \end{aligned}$$

**end**

È facile verificare che tale catena definisce proprio un campionario di Gibbs per la generazione di una  $q$ -colorazione di  $G$  secondo la distribuzione uniforme  $\pi$ .

Inoltre la stessa  $\pi$  è una distribuzione reversibile per la catena di Markov sopra descritta. Infatti, denotando con  $P(f, g)$  la probabilità di andare in un passo dallo stato  $f$  allo stato  $g$  (per

ogni coppia di  $q$ -colorazioni distinte  $f, g$  di  $G$ ), abbiamo che

$$P(f, g) = \begin{cases} 0 & \text{se } f \text{ e } g \text{ si differenziano per il valore attribuito ad almeno 2 nodi} \\ \frac{1}{k} \cdot \frac{1}{\#U} & \text{se } f \text{ e } g \text{ si differenziano per il valore attribuito ad un solo nodo } v \end{cases}$$

Di conseguenza, vale l'equazione  $\pi(f)P(f, g) = \pi(g)P(g, f)$  e quindi  $\pi$  è reversibile. Inoltre, si verifica facilmente che  $P(f, f) > 0$  per ogni  $q$ -colorazione  $f$  di  $G$  e questo implica che la matrice di transizione della catena è aperiodica. Non è detto tuttavia che la matrice sia irriducibile, questo dipende dai valori di  $G$  e di  $q$ .

Nella sezione 4.2 illustreremo una versione non omogenea del campionatore di Gibbs appena descritto, nella quale, a ogni passo, il nodo corrente  $v \in V$  non è più scelto a caso ma viene determinato ciclicamente. Più precisamente, posto  $V = \{v_1, v_2, \dots, v_k\}$ , al passo  $n$ -esimo si sceglie il nodo  $v_j$  di indice  $j = 1 + [n - 1]_k$ . In questo modo, ad ogni sequenza di  $k$  passi, si ridefinisce in modo casuale il colore da attribuire a tutti i nodi del grafo. Mostreremo nel seguito che, nel caso  $q > 2d^2$ , la catena così ottenuta è ergodica.

### 3.4 L'algoritmo di Metropolis

Un altro procedimento classico basato su catene di Markov per generare a caso un elemento secondo una distribuzione fissata è dato dall'algoritmo di Metropolis. Il metodo è molto generale e può essere interpretato come una estensione dei procedimenti di visita casuale dei grafi illustrati nella sezione 1.2.

Consideriamo un insieme finito  $S$  e una distribuzione di probabilità  $\pi$  su  $S$  tale che  $\pi(i) > 0$  per ogni  $i \in S$ . Per generare a caso un elemento in  $S$  secondo la distribuzione  $\pi$ , definiamo innanzitutto un grafo non orientato  $G = (S, E)$  che goda delle seguenti proprietà:

1.  $S$  sia l'insieme dei nodi di  $G$ ,
2.  $G$  sia connesso,
3. il grado di  $G$  (cioè il massimo grado dei suoi nodi) sia limitato da una opportuna costante.

Se denotiamo con  $d_i$  il grado del generico nodo  $i \in S$ , possiamo definire la probabilità  $P(i, j)$  di passare da un nodo  $i$  a un nodo  $j$  nel modo seguente:

$$P(i, j) = \begin{cases} 0 & \text{se } i \neq j \text{ e } \{i, j\} \notin E \\ \frac{1}{d_i} \cdot \min \left\{ \frac{\pi_j d_i}{\pi_i d_j}, 1 \right\} & \text{se } \{i, j\} \in E \\ 1 - \frac{1}{d_i} \cdot \sum_{\ell: \{i, \ell\} \in E} \min \left\{ \frac{\pi_\ell d_i}{\pi_i d_\ell}, 1 \right\} & \text{se } i = j \end{cases}$$

La matrice  $P = [P(i, j)_{i, j \in S}]$  è quindi la matrice di transizione della catena di Markov. Nella simulazione della catena, il generico passo dallo stato corrente  $i$  allo stato successivo  $j$ , può essere eseguito (con probabilità  $P(i, j)$ ) applicando la seguente procedura:

**begin**

scegli a caso  $\ell \in S$  tale che  $\{i, \ell\} \in E$  in modo uniforme

**if**  $\pi_\ell d_i \geq \pi_i d_\ell$  **then**  $j := \ell$

**else** scegli a caso  $j \in \{\ell, i\}$  con le seguenti probabilita':

$$P(\ell) = \frac{\pi_\ell d_i}{\pi_i d_\ell}$$

$$P(i) = 1 - \frac{\pi_\ell d_i}{\pi_i d_\ell}$$

end

Per tale catena valgono inoltre le seguenti proprietà.

1. Poiché  $G$  è connesso la catena è irriducibile.
2. La catena è aperiodica se esistono  $i, j \in S$  tali che  $\pi_j d_i < \pi_i d_j$ , perché questa condizione garantisce un cappio su  $i$ . Un'altra condizione che implica l'aperiodicità è l'esistenza di un ciclo di lunghezza dispari nel grafo  $G$ .
3. La distribuzione  $\pi$  è reversibile per la catena. Infatti, la relazione

$$\pi_i P(i, j) = \pi_j P(j, i)$$

è chiaramente vera quando  $i = j$  e quando  $\{i, j\} \notin E$  e  $i \neq j$ . Inoltre si dimostra facilmente che la stessa vale nel caso  $\{i, j\} \in E$ . Quindi  $\pi$  è sempre la distribuzione stazionaria della catena.

È bene osservare che non sempre la catena è aperiodica. Per esempio, se per ogni  $\{i, j\} \in E$  vale  $\pi_i d_j = \pi_j d_i$  allora la condizione indicata nel punto 2 non è vera e l'aperiodicità non è garantita. Questo capita per esempio nelle passeggiate a caso in un grafo quando non vi sono cicli di lunghezza dispari e la probabilità di passare da un nodo  $i$  a un nodo  $j$  è data da  $P(i, j) = 1/d_i$ .

In ogni caso, se l'aperiodicità vale, allora la catena è ergodica con  $\pi$  distribuzione stazionaria. Possiamo quindi adottare il solito criterio: simuliamo la catena a partire da uno stato qualsiasi per un numero sufficientemente grande di passi e restituiamo il valore dello stato raggiunto. Al crescere del numero di passi sappiamo che la probabilità di restituire un valore  $i$  approssima  $\pi_i$ , per ogni  $i \in S$ .

## Capitolo 4

# Analisi della velocità di convergenza

Un problema naturale che finora abbiamo trascurato è quello di valutare la velocità con la quale le distribuzioni di probabilità di una catena di Markov ergodica convergono alla distribuzione stazionaria.

Se infatti  $\{X_n\}$  è una catena di Markov ergodica e  $\pi$  è la sua distribuzione stazionaria, sappiamo che, qualunque sia lo stato di partenza, la distribuzione di probabilità della variabile  $X_n$  approssima  $\pi$  al crescere di  $n$ . Tuttavia, in generale le due distribuzioni di probabilità non sono uguali, ed è quindi lecito chiedersi quanto deve essere grande  $n$  affinché la loro differenza sia trascurabile.

Più precisamente, fissato un errore  $\varepsilon > 0$  che siamo disposti ad accettare, vogliamo determinare un  $g \in \mathbb{N}$ , dipendente da  $\varepsilon$  e dalla matrice di transizione della catena, tale che per ogni  $n \geq g$  si verifichi

$$|\Pr_i(X_n = j) - \pi(j)| \leq \varepsilon$$

per tutti gli stati  $i, j$ .

Per valutare la differenza tra due distribuzioni di probabilità definite sullo stesso insieme si usa tradizionalmente la seguente nozione di distanza. Siano  $\mu$  e  $\nu$  due distribuzioni di probabilità definite sullo stesso insieme finito  $S$ . Allora la *distanza di variazione totale* tra  $\mu$  e  $\nu$  è data da

$$d_{TV}(\mu, \nu) = \frac{1}{2} \sum_{i \in S} |\mu(i) - \nu(i)| = \frac{1}{2} \|\mu - \nu\|_1$$

È facile verificare che  $d_{TV}$  è una distanza in senso tradizionale. Quindi, data una sequenza di distribuzioni  $\{\mu_n\}_{n \in \mathbb{N}}$  definite su  $S$ , abbiamo che  $\mu_n$  converge a  $\nu$  per  $n \rightarrow +\infty$  se

$$\lim_{n \rightarrow +\infty} d_{TV}(\mu_n, \nu) = 0$$

Tale distanza gode di due proprietà rilevanti:

1)  $d_{TV}(\mu, \nu) = \max_{A \subseteq S} |\mu(A) - \nu(A)|$ , ovvero il suo valore è la massima differenza tra le due distribuzioni valutate su un qualunque sottoinsieme di  $S$ .

2) Se  $X$  e  $Y$  sono due variabili aleatorie definite sullo stesso spazio di probabilità a valori in  $S$  e hanno distribuzione  $\mu$  e  $\nu$ , rispettivamente, allora

$$d_{TV}(\mu, \nu) \leq \Pr(X \neq Y)$$

La dimostrazione di questo fatto è simile ad altre prove svolte nei capitoli precedenti (si veda per esempio la proposizione ??).

## 4.1 Approssimazione generale alla distribuzione stazionaria

La valutazione della velocità di convergenza alla distribuzione stazionaria nel caso generale può essere svolta usando la seguente nozione di coefficiente ergodico.

Data una matrice stocastica  $P \in [0, 1]^{k \times k}$ ,  $P = [p(i, j)]_{i, j=1, \dots, k}$ , chiamiamo *coefficiente ergodico* di  $P$  il valore  $m(P)$  definito da

$$m(P) = \frac{1}{2} \max \left\{ \sum_{u=1}^k |p(i, u) - p(j, u)| \mid i, j = 1, 2, \dots, k \right\}$$

Questo valore misura la differenza tra le righe della matrice  $P$  (che sono vettori stocastici e definiscono quindi distribuzioni di probabilità su  $\{1, 2, \dots, k\}$ ). In particolare valgono le seguenti proprietà la cui verifica è immediata, oppure si può trovare in [9] :

1. Per ogni matrice stocastica  $P$ ,  $0 \leq m(P) \leq 1$ .
2.  $m(P) = 0$  se e solo se  $P$  è stazionaria.
3.  $m(P) = 1$  se e solo se  $P$  ammette due righe con differenza massima, ovvero esistono due indici  $i, j$  tali che non esiste alcun indice  $u$  per il quale  $p(i, u) > 0$  e  $p(j, u) > 0$ .
4.  $m(P) = \max\{d_{TV}(P_i, P_j) \mid i, j \in \{1, 2, \dots, k\}\}$ , dove  $P_i$  e  $P_j$  rappresentano rispettivamente la  $i$ -esima e la  $j$ -esima riga di  $P$ .
5. Se  $P > 0$  allora  $m(P) < 1$ .
6. Per ogni coppia di matrici stocastiche  $A, B$  di ugual dimensione, vale  $m(AB) \leq m(A)m(B)$ .
7. Per ogni matrice stocastica  $P$  e ogni  $n \in \mathbb{N}$ , abbiamo  $m(P^n) \leq (m(P))^n$ .

Tale coefficiente permette di valutare la velocità di convergenza nelle catene primitive. Vale in particolare il seguente risultato di cui omettiamo la dimostrazione [9]. Nota che l'enunciato implica l'ergodicità della catena considerata.

**Teorema 4.1** *Sia  $P$  una matrice stocastica primitiva di dimensione  $k \times k$  e sia  $t$  il minimo intero in  $\mathbb{N}$  tale che  $P^t > 0$ . Se  $\pi$  è la distribuzione stazionaria di  $P$  allora per ogni  $n > t$  e ogni  $i, j \in \{1, 2, \dots, k\}$  vale*

$$|(P^n)_{ij} - \pi_j| \leq [m(P^t)]^{\frac{n}{t}-1}$$

Si può in realtà provare che lo stesso risultato vale per ogni matrice stocastica  $P$  che ammette un  $t \in \mathbb{N}$  tale che  $m(P^t) < 1$ .

**Corollario 4.2** *Nelle stesse ipotesi precedenti, per ogni  $\varepsilon > 0$  e ogni  $i, j \in \{1, 2, \dots, k\}$  si verifica che  $|(P^n)_{ij} - \pi_j| \leq \varepsilon$  per ogni  $n$  tale che*

$$n \geq t \left( 1 + \frac{\log \varepsilon}{\log m(P^t)} \right)$$

La funzione  $g(P, \varepsilon) = t \left( 1 + \frac{\log \varepsilon}{\log m(P^t)} \right)$  rappresenta quindi la velocità di convergenza della catena. Nota che entrambi i valori  $\log \varepsilon$  e  $\log m(P^t)$  sono in generale valori negativi. Inoltre è chiaro che per  $\varepsilon$  decrescente, il valore  $g(P, \varepsilon)$  cresce. Nota anche che il valore  $t$  in molti casi può essere vicino al numero di stati, perché rappresenta la lunghezza minima di cammino necessaria a collegare tutti le coppie di stati. Questo fatto rende la formula poco utile quando il numero degli stati della catena è elevato. Viceversa la formula è utile quando  $t$  è piccolo rispetto al numero di stati, per esempio nel caso in cui tutte le coppie di nodi sono collegabili con cammini di lunghezza limitata. Questo si verifica intuitivamente se il grafo di connessione della catena è abbastanza ricco di lati, ovvero la matrice  $P$  contiene “pochi” zeri. Osserviamo che valori tradizionali di  $\varepsilon$  che consentono una buona approssimazione alla distribuzione stazionaria sono dati da  $\varepsilon = 10^{-3}$  o  $\varepsilon = 10^{-4}$ .

## 4.2 Velocità di convergenza della colorazione di grafi

Valutiamo ora la velocità di convergenza del campionatore di Gibbs descritto nella sezione 3.3. La versione di campionatore che consideriamo tuttavia, è una versione ciclica, nella quale cioè tutti i nodi vengono ricoloreti dopo un numero fissato di passi. Per definire la procedura, ricordiamo anzitutto i dati del problema.

Dato un grafo non orientato  $G = (V, E)$  di  $k$  nodi  $V = \{v_1, v_2, \dots, v_k\}$  e un insieme di colori  $Q = \{1, 2, \dots, q\}$ , sia  $S = \{f \in Q^V \mid f(u) \neq f(v) \forall \{u, v\} \in E\}$  l'insieme delle  $q$ -colorazioni di  $G$  e denotiamo con  $Z_{G,q}$  il loro numero,  $Z_{G,q} = \#S$ . Denotiamo inoltre con  $d$  il *grado* di  $G$ , cioè il massimo tra i gradi dei suoi nodi. Supponiamo che il numero di colori sia abbastanza grande, per esempio  $q > d$ . In questo modo siamo sicuri che  $S \neq \emptyset$  ed è facile determinare una  $q$ -colorazione di  $G$ . Vogliamo generare a caso un elemento di  $S$  secondo una distribuzione di probabilità che approssimi la distribuzione uniforme, definita dalla funzione  $\pi$  tale che

$$\text{per ogni } f \in Q^V, \quad \pi(f) = \begin{cases} \frac{1}{Z_{G,q}} & \text{se } f \in S \\ 0 & \text{altrimenti} \end{cases}$$

La procedura definisce una catena di Markov non omogenea  $\{X_n\}$  sull'insieme degli stati  $S$ . Di fatto al primo passo si sceglie un elemento  $X_0 \in S$  qualsiasi. Quindi si eseguono  $n$  passi nella catena, all' $i$ -esimo passo si ricolore il nodo  $v_j$  dove  $j = 1 + [i - 1]_k$ , scegliendo il nuovo colore in modo uniforme sull'insieme dei colori compatibili con quelli dei nodi adiacenti a  $v_j$ . Formalmente l'algoritmo definito nel modo seguente:

**begin**

  calcola una  $q$ -colorazione  $f \in S$  qualsiasi

**for**  $i = 1, 2, \dots, n$  **do**

**begin**

$j := 1 + [i - 1]_k$

$v := v_j$

$U = \{c \in Q : f(w) \neq c \text{ per ogni } w \in V \text{ tale che } \{w, v\} \in E\}$

      scegli a caso  $c \in U$  secondo la distribuzione uniforme

$f(v) := c$  (lasciando inalterati gli altri valori di  $f(w)$  con  $w \neq v$ )

**end**

**return**  $f$

end

Chiaramente la catena di Markov  $\{X_n\}$  definita implicitamente dalla procedura precedente non è omogenea; piuttosto essa è definita da  $k$  matrici stocastiche  $P_1, P_2, \dots, P_k$ , dove ciascuna  $P_j$  è associata al nodo  $v_j$ . È facile provare, ragionando come nella sezione 3.3, che  $\pi$  è reversibile per tutte le matrici  $P_j$ ,  $j = 1, 2, \dots, k$ , e quindi  $\pi$  rimane distribuzione stazionaria della catena.

Studiamo ora la velocità di convergenza del campionatore appena descritto nell'ipotesi ulteriore che  $q > 2d^2$ , seguendo essenzialmente la presentazione svolta in [7] che consente di semplificare notevolmente la prova.

**Teorema 4.3** *Dato un insieme di colori  $Q = \{1, 2, \dots, q\}$ , sia  $G = (V, E)$  un grafo non orientato di  $k$  nodi, sia  $d$  il grado di  $G$  e sia  $\{X_n\}$  la catena di Markov descritta sopra sulle  $q$ -colorazioni di  $G$ . Denotiamo inoltre con  $\mu^{(n)}$  la distribuzione di probabilità di  $X_n$  (per ciascun  $n \in \mathbb{N}$ ) e con  $\pi$  la distribuzione uniforme sull'insieme delle  $q$ -colorazioni di  $G$ . Se  $q > 2d^2$ , allora esiste una costante  $C > 0$ , dipendente solo da  $d$  e da  $q$ , tale che per ogni  $\varepsilon > 0$  vale*

$$d_{TV}(\mu^{(n)}, \pi) \leq \varepsilon$$

per tutti gli  $n \in \mathbb{N}$  che soddisfano

$$n > Ck(\log k + \log(1/\varepsilon))$$

Prima di illustrare la dimostrazione, elenchiamo le seguenti osservazioni:

1. Nota che il teorema garantisce l'ergodicità di una catena non uniforme, provando che, qualunque sia lo stato iniziale,  $\mu^{(n)}$  converge a  $\pi$ .
2. Il tempo di calcolo richiesto per ottenere una approssimazione con errore minore o uguale a  $\varepsilon$  è dell'ordine  $O(k \log k + k \log(1/\varepsilon))$  poichè, con  $d$  e  $q$  costanti, è chiaro che un passo della catena richiede al più tempo costante. Osserva che questo valore è molto buono perché dipende dal numero di nodi del grafo e non dal numero di stati della catena, che in questo caso potrebbe essere troppo elevato (coincide infatti con il numero di  $q$ -colorazioni del grafo).
3. La condizione  $q > 2d^2$  può in realtà essere rilassata; in effetti il risultato originale ottenuto in [11] richiede solo la condizione  $q > 2d$ , la prova è tuttavia più complicata e poco adatta ad una presentazione in questa sede.
4. Nella dimostrazione si mostrerà un'espressione precisa della costante  $C$  che quindi potrà essere utilizzata nelle applicazioni.

*Dimostrazione.* La prova è basata di nuovo sulla tecnica di accoppiamento. Vogliamo quindi definire una nuova catena di Markov  $\{Y_n\}$  sullo stesso insieme di stati  $S$ , con le stesse matrici di transizione di  $\{X_n\}$  ma con distribuzione iniziale  $\pi$ . Di conseguenza avremo, per ogni  $n \in \mathbb{N}$ ,

$$\Pr(Y_n = f) = \pi(f), \quad \forall f \in S$$

e quindi, ricordando che  $d_{TV}(\mu^{(n)}, \pi) \leq \Pr(X_n \neq Y_n)$ , il nostro problema sarà ridotto a maggiore la probabilità che  $X_n$  e  $Y_n$  siano diversi. Chiaramente dobbiamo fare in modo che le due

variabili  $X_n$  e  $Y_n$  siano definite sul medesimo spazio di probabilità. Notiamo che non è necessario che esse siano indipendenti.

Al generico passo  $n$ -esimo, note le funzioni  $X_{n-1}$  e  $Y_{n-1}$ , consideriamo il nodo corrente  $v = v_{1+[n-1]_k}$ . I nuovi valori  $X_n(v)$  e  $Y_n(v)$  si possono ottenere nel modo seguente: si genera una permutazione casuale uniforme

$$(c_1^n, c_2^n, \dots, c_q^n)$$

degli elementi di  $Q$ ; si assegna a  $X_n(v)$  il primo colore  $c_i^n$  diverso dai colori  $X_{n-1}(w)$  per tutti i  $w$  adiacenti a  $v$ ; in modo analogo, sempre usando la stessa permutazione, si sceglie il colore  $Y_n(v)$ . Formalmente, abbiamo

$$X_n(v) = c_i^n \quad \text{dove } i = \min\{j : c_j^n \neq X_{n-1}(w) \text{ per ogni } w \text{ tale che } \{v, w\} \in E\}$$

$$Y_n(v) = c_{i'}^n \quad \text{dove } i' = \min\{j : c_j^n \neq Y_{n-1}(w) \text{ per ogni } w \text{ tale che } \{v, w\} \in E\}$$

Osserva anche che, se per qualche  $\ell$  vale  $X_\ell = Y_\ell$ , allora  $X_n = Y_n$  per tutti gli  $n \geq \ell$ .

A questo punto vogliamo valutare  $\Pr(X_n(v) \neq Y_n(v))$ , dove  $v$  è il nodo corrente relativo al passo  $n$ -esimo. In seguito chiameremo “fallimento” l’evento  $(X_n(v) \neq Y_n(v))$ . Consideriamo inizialmente il primo ciclo, ovvero quando  $n \leq k$ . Possiamo suddividere l’insieme dei colori in tre sottoinsiemi con le seguenti cardinalità:

1.  $B_0$  sia il numero di colori in  $Q$  diversi da  $X_{n-1}(w)$  e  $Y_{n-1}(w)$  per tutti i nodi  $w$  adiacenti a  $v$

$$B_0 = \#\{c \in Q \mid X_{n-1}(w) \neq c \neq Y_{n-1}(w) \forall w : \{v, w\} \in E\}$$

2.  $B_1$  sia il numero di colori assegnati ai nodi adiacenti a  $v$  in una sola delle due colorazioni  $X_{n-1}$  e  $Y_{n-1}$ .
3.  $B_2$  sia il numero di colori assegnati ai nodi adiacenti a  $v$  in entrambe le colorazioni  $X_{n-1}$  e  $Y_{n-1}$ .

Chiaramente  $q = B_0 + B_1 + B_2$  e inoltre è facile verificare che

$$\Pr(X_n(v) \neq Y_n(v)) = \frac{B_1}{B_0 + B_1}$$

Poiché  $B_2 \leq d$  e  $B_1 \leq 2d - 2B_2$  otteniamo

$$\Pr(X_n(v) \neq Y_n(v)) \leq \frac{2d - 2B_2}{q - B_2} \leq \frac{2d}{q}$$

Consideriamo ora il secondo ciclo:  $k < n \leq 2k$ . Per semplicità, se  $v$  è il nodo corrente al passo  $n$ -esimo, chiamiamo “discrepanza” l’evento  $(\exists w \text{ adiacente a } v \text{ tale che } X_n(w) \neq Y_n(w))$ . Dalla disuguaglianza precedente otteniamo

$$\Pr(\text{discrepanza}) \leq \frac{2d^2}{q}$$

Inoltre, poichè il “fallimento” implica la “discrepanza”, con una prova analoga a quella illustrata per il primo ciclo si dimostra che

$$\Pr(\text{fallimento}/\text{discrepanza}) \leq \frac{2d}{q}$$

e quindi si ottiene

$$\Pr(\text{fallimento}) = \Pr(\text{discrepanza})\Pr(\text{fallimento}/\text{discrepanza}) \leq \frac{2d}{q} \left( \frac{2d^2}{q} \right)$$

Questo implica che per ogni  $v \in V$

$$\Pr(X_{2k}(v) \neq Y_{2k}(v)) \leq \frac{2d}{q} \left( \frac{2d^2}{q} \right)$$

Ragionando nello stesso modo per il terzo ciclo ricaviamo, per ogni  $v \in V$ ,

$$\Pr(X_{3k}(v) \neq Y_{3k}(v)) \leq \frac{2d}{q} \left( \frac{2d^2}{q} \right)^2$$

Le stesse considerazioni applicate all' $m$ -esimo ciclo (per un  $m \in \mathbb{N}$  qualsiasi) portano alla disuguaglianza

$$\Pr(X_{mk}(v) \neq Y_{mk}(v)) \leq \frac{2d}{q} \left( \frac{2d^2}{q} \right)^{m-1}$$

Di conseguenza

$$\Pr(X_{mk} \neq Y_{mk}) \leq \sum_{v \in V} \Pr(X_{mk}(v) \neq Y_{mk}(v)) \leq \frac{k}{d} \left( \frac{2d^2}{q} \right)^m$$

La parte destra dell'ultima maggiorazione è minore o uguale a  $\varepsilon$  se

$$m \geq \frac{\log k - \log \varepsilon - \log d}{\log \frac{q}{2d^2}}$$

Ricordando che all'ultimo passo dell' $m$ -esimo ciclo abbiamo  $n = mk$ , possiamo concludere che

$$d_{TV}(\mu^{(n)}, \pi) \leq \varepsilon$$

per tutti gli interi  $n$  tali che

$$n \geq k \left( 1 + \frac{\log k - \log \varepsilon - \log d}{\log \frac{q}{2d^2}} \right)$$

e questo prova il risultato. □

## Capitolo 5

# Approssimazione per problemi di conteggio

I problemi di conteggio o di enumerazione consistono generalmente nel determinare la cardinalità di un dato insieme di elementi. Molti problemi classici trattati nell'area algoritmica rientrano in questa famiglia e la loro complessità computazionale è stata studiata in letteratura accanto a quella dei tradizionali problemi di decisione [19, 20].

Formalmente un problema di conteggio può essere rappresentato da una funzione  $F : I \rightarrow \mathbb{N}$ , dove  $I$  è l'insieme delle istanze. Esempi tipici sono i seguenti:

1. Dato un grafo orientato  $G$ , due nodi  $u, v$  in  $G$  e un intero  $n > 0$ , determinare il numero di cammini da  $u$  a  $v$  di lunghezza  $n$ .
2. Determinare il numero di alberi di copertura in un grafo non orientato  $G$ .
3. Data una formula booleana  $f(x_1, x_2, \dots, x_n)$ , dove  $x_1, x_2, \dots, x_n$  sono le variabili, determinare il numero di assegnamenti di valori alle variabili che rendono vera la formula.
4. Dato un grafo non orientato  $G$  e un intero  $k$ , determinare il numero di clique di dimensione  $k$  in  $G$ .

Per molti problemi di conteggio sono noti algoritmi esatti efficienti, cioè procedure che su ogni input restituiscono la soluzione esatta e che funzionano in tempo polinomiale. Questo è il caso dei primi due esempi illustrati sopra.

Per altri problemi di conteggio invece algoritmi polinomiali non esistono o non sono stati trovati e si congetta che non sia possibile trovare la soluzione in tempi efficienti. La complessità computazionale dei problemi di enumerazione è stata formalizzata in [20], dove è stata introdotta e studiata la classe  $\#P$  che contiene molti problemi notoriamente difficili per i quali non si conoscono algoritmi che funzionano in tempo polinomiale. Più precisamente, è possibile definire la classe dei problemi  $\#P$ -completi che rappresenta intuitivamente la famiglia dei problemi più difficili in  $\#P$ . Per esempio è noto che i problemi definiti negli esempi 3 e 4 illustrati sopra sono  $\#P$ -completi. Per tali problemi dunque non si conoscono algoritmi di soluzione esatti che funzionano in tempo polinomiale e la loro esistenza è considerata molto improbabile visto che implicherebbe l'equivalenza  $P = NP$ . Tuttavia è spesso possibile definire degli algoritmi di approssimazione efficienti, ovvero procedure che forniscono una soluzione approssimata in un tempo polinomiale.

Un aspetto importante nell'analisi di queste procedure consiste nella valutazione dell'errore; in generale si cerca di garantire che l'errore relativo sia minore di una certa quantità.

Vi sono due nozioni formali di algoritmo di approssimazione, una deterministica e una probabilistica. Entrambe hanno per input una istanza del problema di conteggio e un valore reale  $\varepsilon > 0$  che rappresenta l'errore relativo massimo che il procedimento può compiere. Per questo motivo l'algoritmo è chiamato solitamente *schema di approssimazione*.

Dato un problema di conteggio definito da una funzione  $F : I \rightarrow \mathbb{N}$ , uno schema di approssimazione polinomiale (PTAS) per  $f$  è un algoritmo che riceve in input una coppia  $(x, \varepsilon)$ , dove  $x \in I$  e  $\varepsilon > 0$  è un valore reale, che soddisfa le seguenti condizioni:

1. Per ogni  $\varepsilon > 0$  esiste un polinomio  $p_\varepsilon(y)$  tale che l'algoritmo su input  $(x, \varepsilon)$  richiede un tempo minore o uguale a  $p_\varepsilon(n)$ , dove  $n = |x|$ ;
2. Il valore  $R(x, \varepsilon)$  restituito dall'algoritmo su input  $(x, \varepsilon)$  verifica la relazione

$$(1 - \varepsilon)F(x) \leq R(x, \varepsilon) \leq (1 + \varepsilon)F(x)$$

Se inoltre  $p_\varepsilon(y)$  è polinomialmente limitato anche rispetto a  $1/\varepsilon$ , allora l'algoritmo è chiamato schema di approssimazione *pienamente* polinomiale (FPTAS).

Tale schema di approssimazione garantisce quindi sempre un errore relativo minore della quantità  $\varepsilon$  ricevuta in input:

$$\frac{|R(x, \varepsilon) - F(x)|}{F(x)} \leq \varepsilon$$

Tuttavia in molti casi anche questo tipo di approssimazione sembra difficile da stabilire. In particolare, per certi problemi di conteggio si è dimostrato che non esiste un PTAS a meno che la classe  $P$  non coincida con  $NP$ . Questo significa che quasi certamente il problema non è approssimabile mediante un algoritmo deterministico. In questi casi l'unica possibilità per trattare il problema con un algoritmo efficiente è quella di trovare uno schema di approssimazione probabilistico che fornisca un errore relativo che con alta probabilità è minore della quantità  $\varepsilon$  fissata a priori.

Formalmente, uno schema *probabilistico* di approssimazione polinomiale (RPTAS) per una funzione  $F : I \rightarrow \mathbb{N}$ , è un algoritmo probabilistico che su input  $(x, \varepsilon)$  verifica la condizione 1 della definizione precedente e inoltre, su ogni input  $(x, \varepsilon)$ , restituisce il valore  $R(x, \varepsilon)$  tale che

$$\Pr((1 - \varepsilon)F(x) \leq R(x, \varepsilon) \leq (1 + \varepsilon)F(x)) \geq 2/3$$

Anche in questo caso diremo che lo schema è *pienamente* polinomiale (RFPTAS) se  $p_\varepsilon(y)$  è polinomialmente limitato rispetto a  $1/\varepsilon$ . Ovvero, esiste un polinomio  $q(y, z)$  in  $y$  e  $z$ , tale che  $p_\varepsilon(n) \leq q(n, 1/\varepsilon)$ , per ogni  $n \in \mathbb{N}$  e ogni  $\varepsilon > 0$ .

Il valore  $2/3$  nella definizione precedente può essere sostituito da un qualunque numero reale  $\delta$  tale che  $1/2 < \delta < 1$ . Vale infatti la seguente proprietà.

**Proposizione 5.1** *Consideriamo un problema di conteggio definito da una funzione  $F : I \rightarrow \mathbb{N}$  e sia  $\delta$  una costante tale che  $1/2 < \delta < 1$ . Supponiamo che per tale problema vi sia un RPTAS che su ogni input  $(x, \varepsilon)$  restituisce un valore  $R(x, \varepsilon)$  tale che*

$$\Pr(|R(x, \varepsilon) - F(x)| \leq \varepsilon F(x)) \geq \delta \quad .$$

Allora, per ogni  $\alpha > 0$ , esiste un RPTAS per lo stesso problema che su ogni input  $(x, \varepsilon)$  restituisce un valore  $R'(x, \varepsilon)$  tale che

$$\Pr(|R'(x, \varepsilon) - F(x)| > \varepsilon F(x)) < \alpha$$

Per la dimostrazione basta usare la mediana di un certo numero di ripetizioni dell'algoritmo sullo stesso input e applicare la disuguaglianza di Chebychev sulle binomiali.

## 5.1 Conteggio approssimato di colorazioni

Un esempio rilevante di RPTAS per un problema di conteggio difficile riguarda il calcolo del numero di  $q$ -colorazioni. L'algoritmo che consideriamo è stato proposto in [11] e la versione semplificata che qui illustriamo è dovuta a [7]. Tale procedura si basa principalmente sul procedimento di generazione casuale di  $q$ -colorazioni presentato nella sezione 3.3.

Anche in questo caso l'istanza del problema è data da un grafo non orientato  $G = (V, E)$  e da un numero  $q$  di colori. Denotiamo con  $k$  il numero di nodi del grafo, con  $d$  il massimo grado dei suoi vertici e con  $m$  il numero dei suoi lati. Così la coppia  $(k, m)$  rappresenta la dimensione dell'istanza del problema. Inoltre, rappresentiamo con  $Q = \{1, 2, \dots, q\}$  l'insieme dei colori e ricordiamo che una  $q$ -colorazione è una funzione  $f : V \rightarrow Q$  tale che, per ogni  $\{u, v\} \in E$ ,  $f(u) \neq f(v)$ . La soluzione del problema è il numero di  $q$ -colorazioni del grafo, che denotiamo con  $Z_{G,q}$ ; poiché il problema è notoriamente un problema difficile ( $\#P$ -completo) vogliamo definire uno schema di approssimazione probabilistico (RPTAS). Osserviamo che per questo problema non esistono schemi di approssimazione polinomiale deterministici (PTAS). L'unico modo per approssimare la soluzione sembra dunque quella di utilizzare una procedura probabilistica.

Descriviamo ora il procedimento adottato dall'algoritmo. Consideriamo l'insieme dei lati  $E = \{e_1, e_2, \dots, e_m\}$  del grafo e, per ogni  $j = 0, 1, \dots, m$ , definiamo il grafo  $G_j = (V, \{e_1, \dots, e_j\})$  con la convenzione che  $G_0 = (V, \emptyset)$ . Nota che tutti questi grafi hanno lo stesso insieme di vertici  $V$ . Inoltre  $G_m = G$  e ciascun  $G_j$  è ottenuto da  $G_{j-1}$  aggiungendo il lato  $e_j$ ,  $j = 1, 2, \dots, m$ . Possiamo anche denotare con  $Z_j$  il numero di  $q$ -colorazioni di  $G_j$ , per ogni  $j = 0, 1, \dots, m$ . Di conseguenza  $Z_m = Z_{G,q}$  ed è facile verificare che  $Z_0 = q^k$  poiché nel grafo  $G_0$  ogni assegnamento di colori ai nodi è una  $q$ -colorazione. Quindi il valore  $Z_m$  che vogliamo calcolare può essere ottenuto mediante la produttorina seguente:

$$Z_m = Z_0 \cdot \frac{Z_1}{Z_0} \cdot \frac{Z_2}{Z_1} \cdot \dots \cdot \frac{Z_m}{Z_{m-1}} \quad (5.1)$$

Possiamo quindi valutare i rapporti  $\frac{Z_j}{Z_{j-1}}$  e moltiplicare queste frazioni tra loro. Osserva che ciascun  $Z_j$  può essere visto come il numero di  $q$ -colorazioni di  $G_{j-1}$  che assegnano colori diversi ai nodi del lato  $e_j$ . Denotiamo ora con  $x_j$  e  $y_j$  i due nodi del lato  $e_j$ , per ciascun  $j = 1, 2, \dots, m$ . Allora

$$Z_j = \#\{f \in Q^V \mid f \text{ è } q\text{-colorazione di } G_{j-1} \text{ tale che } f(x_j) \neq f(y_j)\}$$

Di conseguenza  $\frac{Z_j}{Z_{j-1}}$  è la probabilità che una  $q$ -colorazione  $f$  di  $G_{j-1}$  estratta con distribuzione uniforme, sia anche una colorazione di  $G_j$ , ovvero

$$\frac{Z_j}{Z_{j-1}} = \Pr(f(x_j) \neq f(y_j))$$

L'idea dell'algoritmo è allora la seguente: per ogni  $j = 1, 2, \dots, m$ , usando il campionatore di Gibbs descritto nella sezione 3.3, si generano diverse  $q$ -colorazioni casuali del grafo  $G_{j-1}$  e si verifica quante di queste attribuiscono colori diversi ai nodi  $x_j$  e  $y_j$ ; si ottiene così una stima  $Y_j$  della probabilità  $\frac{Z_j}{Z_{j-1}}$ . Moltiplicando tutti questi valori tra di loro insieme alla quantità  $Z_0 = q^k$  si ottiene il risultato  $R$  fornito dall'algoritmo

$$R = q^k \cdot \prod_{j=1}^m Y_j$$

che per la relazione (5.1) rappresenta una approssimazione di  $Z_m$ .

Osserviamo che questo procedimento dipende da due parametri importanti, che denotiamo con  $\ell$  e  $n$ , rispettivamente:

- $\ell$  è il numero di  $q$ -colorazioni casuali di  $G_{j-1}$ , generate mediante il campionatore di Gibbs, utilizzate per calcolare ciascuna stima  $Y_j$  della probabilità  $\frac{Z_j}{Z_{j-1}}$ ,  $j = 1, 2, \dots, m$ . Così risulta  $Y_j = H_j/\ell$ , dove  $H_j$  è il numero di  $q$ -colorazioni di  $G_{j-1}$  generate a caso che attribuiscono colori diversi ai nodi  $x_j$  e  $y_j$ .
- $n$  è il numero di passi principali eseguiti da ciascun campionatore di Gibbs per determinare la  $q$ -colorazione casuale corrispondente (ovvero il numero delle transizioni compiute sulla catena di Markov associata).

L'algoritmo è riassunto dalla seguente procedura ad alto livello:

```

begin
  R = qk
  for j = 1, 2, ..., m do
    begin
      % calcola Yj %
      H := 0
      for i = 1, 2, ..., ℓ do
        begin
          % esegui una simulazione del campionatore di Gibbs su Gj-1 %
          f := colorazione iniziale di Gj-1
          for t = 1, 2, ..., n do
            {
              esegui una transizione del campionatore di Gibbs su Gj-1
              sia g la nuova colorazione ottenuta
              f := g
            }
            if f(xj) ≠ f(yj) then H := H + 1
          end
          Yj := H/ℓ
          R := R · Yj
        end
      end
    return R
  end
end

```

Affinché questo procedimento sia davvero un RPTAS per il problema considerato dovremo garantire che l'errore relativo compiuto sia piccolo a piacere con alta probabilità. Questo significa

determinare i valori dei due parametri  $\ell$  e  $n$  per i quali il risultato  $R$  della procedura soddisfi la relazione

$$\Pr(|R - Z_m| > \varepsilon Z_m) \leq 1/3$$

La seguente proposizione afferma proprio che sotto certe ipotesi, riguardanti  $q$  e  $d$ , tali valori esistono e quindi l'algoritmo illustrato è davvero un RPTAS per il nostro problema.

**Teorema 5.2** *Assumendo la notazione illustrata sopra, supponiamo che  $d \geq 2$  e  $q > 2d^2$ . Allora esiste un RPTAS per il problema di determinare il numero di  $q$ -colorazioni di  $G$ .*

Prima di descrivere la dimostrazione, osserviamo che la condizione  $d \geq 2$  è del tutto naturale perché nel caso  $d \leq 1$  il numero di  $q$ -colorazioni si determina facilmente. L'ipotesi  $q > 2d^2$  invece deriva dall'utilizzo del Teorema 4.3. Tuttavia, poiché lo stesso teorema vale nell'ipotesi più debole  $q > 2d$ , anche nel nostro caso quest'ultima condizione è sufficiente a garantire la validità dell'enunciato.

La dimostrazione del teorema 5.2 si basa sui seguenti lemmi.

**Lemma 5.3** *Per ogni  $\varepsilon > 0$  e ogni sequenza  $a_1, a_2, \dots, a_k$  tali che*

$$\left(1 - \frac{\varepsilon}{2k}\right) \leq a_i \leq \left(1 + \frac{\varepsilon}{2k}\right) \quad \text{per ogni } i = 1, \dots, k$$

vale

$$1 - \varepsilon \leq \prod_{i=1}^k a_i \leq 1 + \varepsilon$$

*Dimostrazione.* Ragionando per induzione si prova che per ogni  $i \in \mathbb{N}$

$$\left(1 - \frac{\varepsilon}{2k}\right)^i \geq 1 - \frac{i\varepsilon}{2k}$$

Di conseguenza

$$\prod_{i=1}^k a_i \geq \left(1 - \frac{\varepsilon}{2k}\right)^k \geq 1 - \frac{\varepsilon}{2} \geq 1 - \varepsilon$$

e questo prova la prima disuguaglianza. Per provare la seconda osserva che, per ogni  $0 \leq x \leq 1$ , vale

$$1 + \frac{x}{2} \leq e^{x/2} \leq 1 + x$$

e quindi

$$\prod_{i=1}^k a_i \leq \left(1 + \frac{\varepsilon}{2k}\right)^k \leq e^{\varepsilon/2} \leq 1 + \varepsilon$$

□

**Lemma 5.4** *Supponi  $d \geq 2$  e  $q \geq 2d$  per due interi positivi  $d, q$ . Sia  $G$  un grafo di grado minore o uguale a  $d$  e sia  $f$  una  $q$ -colorazione di  $G$  estratta uniformemente a caso. Allora, per ogni coppia di nodi distinti  $x, y$  di  $G$  abbiamo*

$$\Pr(f(x) \neq f(y)) \geq 1/2$$

*Dimostrazione.* Supponi che  $\{x, y\}$  non sia un lato del grafo (altrimenti la proprietà è ovvia). Osserva che una volta fissati i colori  $f(w)$  dei nodi  $w$  diversi da  $x$ , la variabile aleatoria  $f(x)$  possiede una distribuzione uniforme sull'insieme dei colori disponibili, cioè quelli diversi dai valori  $\{f(z) \mid z \text{ vicino di } x\}$ . I colori disponibili sono almeno  $q - d$  e quindi la probabilità condizionata che  $f(x)$  assuma uno di questi è minore o uguale a  $\frac{1}{q-d} \leq 1/2$ . Sia  $G'$  il sottografo di  $G$  ottenuto eliminando il nodo  $x$  e sia  $g$  una  $q$ -colorazione di  $G'$  fissata. Allora abbiamo

$$\Pr(f(x) = g(y) \mid f(w) = g(w) \forall w \neq x) \begin{cases} = 0 & \text{se } g(y) = g(z) \text{ per qualche } z \text{ vicino di } x \\ \leq \frac{1}{q-d} \leq \frac{1}{2} & \text{altrimenti} \end{cases}$$

Di conseguenza otteniamo

$$\begin{aligned} \Pr(f(x) = f(y)) &= \\ &= \sum_{g \text{ } q\text{-col. di } G'} \Pr(f(x) = g(y) \mid f(w) = g(w) \forall w \neq x) \Pr(f(w) = g(w) \forall w \neq x) \\ &\leq \frac{1}{2} \sum_{g \text{ } q\text{-col. di } G'} \Pr(f(w) = g(w) \forall w \neq x) = \frac{1}{2} \end{aligned}$$

□

**Dimostrazione del teorema 5.2.** Valutiamo anzitutto l'errore relativo compiuto dall'algoritmo. Vogliamo mostrare che per opportuni valori dei parametri  $n$  e  $\ell$  il risultato  $R$  ottenuto su input  $(G, q, \varepsilon)$  soddisfa la relazione

$$\Pr((1 - \varepsilon)Z_m \leq R \leq (1 + \varepsilon)Z_m) \geq 2/3 \quad (5.2)$$

A tale scopo supponi che, per ogni  $j = 1, 2, \dots, m$ , valga la disuguaglianza

$$\left(1 - \frac{\varepsilon}{2m}\right) \frac{Z_j}{Z_{j-1}} \leq Y_j \leq \left(1 + \frac{\varepsilon}{2m}\right) \frac{Z_j}{Z_{j-1}} \quad (5.3)$$

Allora, poiché  $R = q^k \prod_{j=1}^m Y_j$ , per il lemma 5.3, si ottiene esattamente l'evento

$$(1 - \varepsilon)Z_m \leq R \leq (1 + \varepsilon)Z_m$$

di cui vogliamo stimare la probabilità.

Per il lemma 5.4 sappiamo che  $\frac{Z_j}{Z_{j-1}} \geq 1/2$  e quindi la relazione (5.3) è conseguenza delle disuguaglianze

$$-\frac{\varepsilon}{4m} \leq Y_j - \frac{Z_j}{Z_{j-1}} \leq \frac{\varepsilon}{4m} \quad j = 1, 2, \dots, m$$

Quest'ultima può essere spezzata nelle due disuguaglianze seguenti

$$\begin{aligned} (i) \quad & |Y_j - \mu^{(n)}(f(x_j) \neq f(y_j))| \leq \frac{\varepsilon}{8m} \\ (ii) \quad & |\mu^{(n)}(f(x_j) \neq f(y_j)) - \frac{Z_j}{Z_{j-1}}| \leq \frac{\varepsilon}{8m} \end{aligned}$$

dove  $\mu^{(n)}$  è la distribuzione della  $q$ -colorazione generata dal campionatore di Gibbs all' $n$ -esimo passo.

Per dimostrare il teorema dobbiamo quindi provare che le disuguaglianze (i) e (ii) valgono per tutti i  $j = 1, 2, \dots, m$  con probabilità almeno  $2/3$ . La disuguaglianza (i) rappresenta l'errore dovuto alla differenza tra la stima empirica  $Y_j$  ottenuta dalla valutazione di  $\ell$   $q$ -colorazioni generate dal campionatore di Gibbs e il suo valor medio  $\mu^{(n)}(f(x_j) \neq f(y_j))$ . La disuguaglianza (ii) rappresenta invece l'errore compiuto dal campionatore di Gibbs che produce una  $q$ -colorazione con distribuzione  $\mu^{(n)}$  e non con distribuzione uniforme.

Osserviamo subito che per il Teorema 4.3 la (ii) è sempre vera per tutti i  $j$  a patto che  $n$  soddisfi la relazione

$$n \geq k \left( 1 + \frac{\log k + \log \frac{8m}{\varepsilon} - \log d}{\log \frac{q}{2d^2}} \right)$$

Poiché  $m \leq kd$  la relazione precedente è soddisfatta se

$$n \geq k \left( 1 + \frac{2 \log k - \log \varepsilon + \log 8}{\log \frac{q}{2d^2}} \right) \quad (5.4)$$

Per quanto riguarda la (i) invece, osserva che  $H_j = \ell Y_j$  è una binomiale di parametri  $\ell$  e  $p = \mu^{(n)}(f(x_j) \neq f(y_j))$ . Quindi la disuguaglianza (i) equivale a

$$|H_j - \ell p| \leq \frac{\varepsilon \ell}{8m}$$

Applicando ora la disuguaglianza di Chebychev otteniamo

$$\Pr \left( |H_j - \ell p| > \frac{\varepsilon \ell}{8m} \right) \leq \frac{p(1-p)(8m)^2}{\varepsilon^2 \ell} \leq \frac{16m^2}{\varepsilon^2 \ell}$$

Il termine destro è minore o uguale a  $\frac{1}{3m}$  per

$$\ell \geq \frac{48m^3}{\varepsilon^2} \quad (5.5)$$

Di conseguenza per tali valori di  $\ell$  abbiamo che per ogni  $j = 1, 2, \dots, m$

$$\Pr \left( |H_j - \ell p| > \frac{\varepsilon \ell}{8m} \right) \leq \frac{1}{3m}$$

ovvero

$$\Pr \left( \forall j = 1, \dots, m : |Y_j - \mu^{(n)}(f(x_j) \neq f(y_j))| \leq \frac{\varepsilon}{8m} \right) \geq \frac{2}{3}$$

Ciò significa che la probabilità che la (i) valga per tutti i  $j = 1, 2, \dots, m$  è maggiore o uguale a  $2/3$  e, per l'argomentazione precedente, questo implica la (5.2).

Valutiamo infine il tempo di calcolo richiesto dalla procedura per garantire l'errore relativo appena ottenuto. L'algoritmo ripete tre cicli, rispettivamente  $m$ ,  $\ell$  e  $n$  volte. Supponendo costante il grado  $d$  del grafo possiamo assumere che il passo generico del campionatore di Gibbs richieda tempo  $O(1)$ . Di conseguenza, tenendo conto di (5.5) e (5.4), l'algoritmo può essere eseguito in tempo  $O(m\ell n) = O(k^5 \log k)$ .



# Bibliografia

- [1] A.V. Aho, J.E. Hopcroft, J.D. Ullman, *The design and analysis of computer algorithms*, Addison-Wesley, 1974.
- [2] A. Bertoni, M. Goldwurm, *Progetto e analisi di algoritmi*, Rapporto Interno n. 230-98, Dip. Scienze dell'Informazione, Univ. degli Studi di Milano, 1998.
- [3] P. Brémaud, *Markov Chains: Gibbs fields, Monte Carlo Simulation and Queues*, Springer, New York, 1998.
- [4] T.H. Cormen, C.E. Leiserson, R.L. Rivest, *Introduction to algorithms*, The MIT Press, McGraw-Hill, 1990.
- [5] W. Feller, *An Introduction to Probability Theory and Applications*, J. Wiley, 1968.
- [6] P. Flajolet, P. Zimmerman, and B. Van Cutsem. A calculus for the random generation of labelled combinatorial structures. *Theoretical Computer Science*, 132(1-2):1–35, 1994.
- [7] O. Häggström, *Finite Markov Chains and Algorithmic Applications*, London Mathematical Society, 2003.
- [8] J. Hromkovic, *Design and analysis of randomized algorithms*, Springer, 2005.
- [9] M. Iosifescu, *Finite Markov Processes and Their Applications*, J. Wiley and Sons, 1980.
- [10] D. Isaacson, R. Madsen, *Markov Chains Theory and Applications*, R.E. Krieger Publishing Company, 1985.
- [11] M. Jerrum, A very simple algorithm for estimating the number of  $k$ -colorings of a low-degree graph. *Random Structures and Algorithms*, vol. 7, 157-165, 1995.
- [12] M. Jerrum, Mathematical Foundations of the Markov chain Monte Carlo method, in *Probabilistic Methods for Algorithmic discrete Mathematics*, M. Habib, C. McDiarmid, J. Ramirez-Alfonsin and B. Reed editors, Springer, 1998.
- [13] M. Jerrum, A. Sinclair, A Markov chain Monte Carlo method: an approach to approximate counting and integration, in *Approximations for NP-hard problems*, D. Hochbaum editor, PSW Publishing, 1996, 482-520.
- [14] J.G. Kemeny e J.L. Snell, *Finite Markov Chains*, Van Nostrand, 1960.
- [15] A. Paz, *Introduction to Probabilistic Automata*, Academic Press, 1971.

- [16] A. Salomaa e M. Soittola, *Automata-Theoretic Aspects of Formal Power Series*, Springer-Verlag, 1978.
- [17] E. Seneta, *Non-negative Matrices and Markov Chains*, Springer-Verlag, 1981.
- [18] G. Strang, *Linear Algebra and its Applications*, Harcourt Brace & Company, 1988.
- [19] L.G. Valiant, The complexity of computing the permanent, *Theoret. Comput. Sci.*, vol. 8, 189-201, 1979.
- [20] L.G. Valiant, The complexity of enumeration and reliability problems, *SIAM J. Comput.*, vol. 8, 410-421, 1979.
- [21] W. Woess, *Catene di Markov e teoria del potenziale nel discreto*, Quaderni dell'Unione Matematica Italiana, n.41, Pitagora Editrice 1996.