

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

Dipartimento di Fisica e Astronomia “Augusto Righi”
Corso di Laurea in Fisica

CATENE DI MARKOV

Relatore:
Prof. Nicola Arcozzi

Presentata da:
Maddalena Montanari

Anno Accademico 2024/2025

Abstract

In questa tesi vengono introdotti i concetti fondamentali relativi alle catene di Markov e se ne studia il comportamento attraverso simulazioni numeriche.

Il lavoro è organizzato in due capitoli teorici e due dedicati all'analisi sperimentale.

Inizialmente vengono definiti alcuni dei concetti principali appartenenti alla teoria della probabilità, in particolare quelli legati alle variabili aleatorie discrete.

Il secondo capitolo è invece dedicato interamente alle catene di Markov; dopo la definizione, vengono introdotti concetti chiave come stati, probabilità di transizione e matrice di transizione.

Vengono successivamente introdotte alcune proprietà rilevanti, tra cui la distribuzione stazionaria, che descrive la probabilità di permanenza a lungo termine in ciascuno stato, e il comportamento asintotico, che permette di comprendere l'evoluzione della catena nel tempo e il raggiungimento di un equilibrio.

I capitoli sperimentali fanno uso di simulazioni numeriche per analizzare alcune proprietà delle catene di Markov.

Usando il linguaggio di programmazione Python sono state generate passeggiate aleatorie, con l'obiettivo di simulare il comportamento dei processi di Markov. In particolare, è stato possibile osservare l'evoluzione del sistema nel tempo.

Sono state considerate classi di grafi generati casualmente, corrispondenti a differenti configurazioni dello spazio degli stati. Tali grafi permettono di studiare l'effetto della struttura della rete sulle proprietà delle catene di Markov.

Indice

1	Probabilità	5
1.1	Definizione di probabilità	5
1.1.1	Tribù/ σ -algebra	5
1.1.2	Definizione di probabilità - Kolmogorov	6
1.2	Indipendenza	6
1.3	Variabili aleatorie discrete	7
1.3.1	Valore medio di una variabile aleatoria discreta	8
1.3.2	Varianza di una variabile aleatoria discreta	8
1.4	Grafo aleatorio	8
2	Catene di Markov	9
2.1	Catene di Markov e probabilità di transizione	9
2.1.1	Definizione	9
2.1.2	Esempio: random walk	10
2.1.3	Proprietà di Markov	10
2.1.4	Matrice di transizione	11
2.1.5	Le equazioni di Chapman-Kolmogorov	11
2.1.6	Evoluzione della distribuzione e stazionarietà	12
2.1.7	Accessibilità e classi di comunicazione	12
2.1.8	Periodo	13
2.2	Catene di Markov a spazio di stati finito	13
2.3	Tempi di arresto, proprietà di Markov forte	13
2.3.1	Tempi di arresto	13
2.3.2	Proprietà di Markov forte	14
2.4	Ricorrenza e transienza	14
2.4.1	Definizioni	14
2.4.2	Teorema di Polya	15
2.5	Catena delle escursioni	15
2.5.1	Esempio di catena delle escursioni	16
2.5.2	Proiezione sull'ultima lettera	16
2.5.3	Distribuzione stazionaria della catena delle escursioni	17

2.6	Escursioni e legge forte dei grandi numeri	18
2.6.1	Definizione: estensione additiva	18
2.6.2	Esempio: catena con 3 stati	19
2.6.3	Equilibrio asintotico	19
2.6.4	Teorema di sintesi	20
2.7	Accoppiamento e Teorema del limite di Kolmogorov	20
3	Simulazione in python di passeggiate aleatorie dentro labirinti	22
3.1	Catene di Markov nei labirinti	22
3.1.1	Classificazione degli stati	23
3.1.2	Struttura	23
3.2	Generazione del labirinto	24
3.3	Obiettivi	25
3.4	Simulazione random walk nel labirinto	26
3.5	Rappresentazione grafica del labirinto e tracciamento del percorso	27
3.6	Probabilità di uscita e numero medio di passi	28
3.7	Distribuzione stazionaria e varianza	29
3.8	Confronto fra diverse configurazioni di labirinto	30
3.8.1	Analisi al variare della dimensione	30
3.8.2	Analisi al variare della densità e confronto con random walk classico 2D	31
3.8.3	Specchi riflettenti	32
3.9	Cosa accadrebbe se venisse violata la proprietà di Markov?	32
4	Risultati della simulazione e conclusione	35
4.1	Analisi al variare della dimensione	35
4.1.1	Dati numerici e grafici	35
4.1.2	Discussione dei risultati	37
4.2	Analisi al variare della densità	38
4.2.1	Dati numerici e grafici	38
4.2.2	Discussione dei risultati	40
4.3	Confronto con random walk classico 2D	40
4.3.1	Dati numerici e grafici	40
4.3.2	Discussione dei risultati	42
4.4	Specchi riflettenti	43
4.4.1	Dati numerici e grafici	43
4.4.2	Discussione dei risultati	45
4.5	Violazione della proprietà di Markov	45
4.5.1	Dati numerici e grafici	45
4.5.2	Discussione dei risultati	46
4.6	Conclusioni	46

A Dimostrazione Teorema sezione 2.3	48
B Link al codice Python	53

Capitolo 1

Probabilità

In questo capitolo si richiamano i concetti fondamentali della teoria della probabilità. Vengono introdotte la definizione di probabilità, la nozione di indipendenza e il concetto di variabile aleatoria discreta, insieme alle principali grandezze ad essa associate.

Infine, viene data la definizione di grafo aleatorio.

Il contenuto di questo capitolo è basato principalmente su [1].

1.1 Definizione di probabilità

La prima definizione di probabilità è dovuta a Laplace all'inizio dell'Ottocento ed è riassunta nella formula

$$P(A) = \frac{N(A)}{N(\Omega)}. \quad (1.1)$$

Dove Ω indica lo spazio campionario, l'insieme dei risultati possibili.

Essa si interpreta tradizionalmente dicendo che la probabilità di un evento A è il rapporto tra il numero $N(A)$ dei casi favorevoli al realizzarsi dell'evento A e il numero $N(\Omega)$ dei casi possibili, ammesso siano ugualmente probabili.

Questa definizione è intuitiva e utile in molte situazioni semplici, ma presenta limiti importanti: vale solo per spazi finiti, richiede l'ipotesi non sempre giustificabile di equiprobabilità e, soprattutto, si fonda implicitamente sul concetto di probabilità che vorrebbe definire.

1.1.1 Tribù/ σ -algebra

Definizione 1. Si chiama *tribù*, o *σ -algebra*, una famiglia \mathcal{F} di sottoinsiemi di Ω ,

$$\mathcal{F} \subset \mathcal{P}(\Omega),$$

che goda delle seguenti proprietà:

1. $\Omega \in \mathcal{F}$;
2. $A \in \mathcal{F} \Rightarrow A^c \in \mathcal{F}$;
3. $\forall n \in \mathbb{N}, A_n \in \mathcal{F} \Rightarrow \bigcup_{n \in \mathbb{N}} A_n \in \mathcal{F}$.

Una tribù è stabile rispetto all'operazione di unione numerabile.

Dalla definizione e dalle leggi di De Morgan segue che una σ -algebra è stabile anche rispetto a unioni finite e intersezioni (finite o numerabili).

1.1.2 Definizione di probabilità - Kolmogorov

Definizione 2. Sia Ω un insieme non vuoto, detto **spazio campionario**, e sia \mathcal{F} una tribù (o σ -algebra) di sottoinsiemi di Ω .

Una **probabilità** su (Ω, \mathcal{F}) è un'applicazione

$$P : \mathcal{F} \longrightarrow [0, 1]$$

tale che valgano i seguenti assiomi:

1. $P(\Omega) = 1$;
2. $P(A) \geq 0$ per ogni $A \in \mathcal{F}$;
3. per ogni successione numerabile di insiemi a due a due disgiunti $\{A_n\}_{n \in \mathbb{N}} \subset \mathcal{F}$ vale

$$P\left(\bigcup_{n=1}^{\infty} A_n\right) = \sum_{n=1}^{\infty} P(A_n).$$

La terna (Ω, \mathcal{F}, P) si dice **spazio di probabilità**.

Questa definizione fornisce una base rigorosa per il calcolo delle probabilità e consente di trattare in modo unificato fenomeni discreti e continui.

1.2 Indipendenza

Definizione 3. Due eventi A e B si dicono (stocasticamente) **indipendenti** rispetto alla probabilità P se vale

$$P(A \cap B) = P(A)P(B). \quad (1.2)$$

La nozione di indipendenza è legata alla particolare misura di probabilità P . Due eventi indipendenti rispetto a una probabilità possono non esserlo rispetto a un'altra probabilità.

Il concetto di indipendenza può essere esteso all'indipendenza di tribù:

Definizione 4. Sia P una probabilità definita su \mathcal{F} , tribù di sottoinsiemi di Ω , e siano $\mathcal{F}_1, \dots, \mathcal{F}_n$ tribù contenute in \mathcal{F} . Esse si dicono **indipendenti** se, per ogni scelta di eventi

$$A_i \in \mathcal{F}_i \quad (i = 1, \dots, n),$$

vale

$$P\left(\bigcap_{i=1}^n A_i\right) = \prod_{i=1}^n P(A_i). \quad (1.3)$$

Le ultime due definizioni sono legate dal seguente teorema:

Definizione 5. Due eventi A e B sono indipendenti se e solo se le tribù $\mathcal{F}(A)$ e $\mathcal{F}(B)$ che essi generano sono indipendenti.

Generalizzando:

Definizione 6. Si dice che n eventi A_1, \dots, A_n sono indipendenti se lo sono le tribù $\mathcal{F}(A_1), \dots, \mathcal{F}(A_n)$ che essi generano. Gli eventi di una successione $(A_n)_n \in \mathbb{N}$ si dicono indipendenti se, per ogni $n \in \mathbb{N}$, gli eventi A_1, A_2, \dots, A_n sono indipendenti.

Teorema 7. Dati n eventi A_1, A_2, \dots, A_n sono equivalenti le seguenti condizioni:

1. essi sono indipendenti;
2. per ogni numero naturale $k \leq n$ e per ogni scelta di k indici $i_1, \dots, i_k \in \{1, \dots, n\}$ si ha

$$P\left(\bigcap_{j=1}^k A_{i_j}\right) = \prod_{j=1}^k P(A_{i_j}).$$

1.3 Variabili aleatorie discrete

Definizione 8. Dato uno spazio misurabile (Ω, \mathcal{F}) , si chiama **variabile aleatoria discreta** ogni funzione X definita in Ω che assuma un numero finito o un'infinità numerabile di valori $\{x_n\}$ e tale che, per ogni valore x_n , il sottoinsieme di Ω nel quale X assume il valore x_n ,

$$\{X = x_n\} = X^{-1}(\{x_n\}),$$

sia misurabile, cioè appartenga alla tribù \mathcal{F} .

Se poi Ω è esso stesso finito o numerabile, una variabile aleatoria discreta sarà una qualsiasi funzione a valori in \mathbb{R} definita in Ω .

1.3.1 Valore medio di una variabile aleatoria discreta

Definizione 9. Nello spazio di probabilità (Ω, \mathcal{F}, P) , sia X una variabile aleatoria discreta. Se la serie

$$\sum_j |x_j| P(X = x_j)$$

converge, si dice **speranza matematica**, o **valore medio**, o semplicemente **media** di X la somma

$$E(X) = E\left(\sum_i x_i \mathbf{1}_{A_i}\right) = \sum_i x_i P(A_i) = \sum_j x_j P(X = x_j). \quad (1.4)$$

1.3.2 Varianza di una variabile aleatoria discreta

Definizione 10. Nello spazio di probabilità (Ω, \mathcal{F}, P) , sia X una variabile aleatoria discreta. Se la serie

$$\sum_j x_j^2 P(X = x_j)$$

converge, si definisce **varianza** di X la quantità

$$\text{Var}(X) = E[(X - E(X))^2] = \sum_j (x_j - E(X))^2 P(X = x_j) \quad (1.5)$$

Vale anche la formula equivalente

$$\text{Var}(X) = E(X^2) - [E(X)]^2 \quad (1.6)$$

La varianza misura la dispersione dei valori della variabile aleatoria rispetto alla sua media.

1.4 Grafo aleatorio

Un grafo aleatorio è un grafo la cui struttura è determinata da un processo probabilistico.

In particolare, dato un insieme di vertici V , gli archi del grafo sono determinati da variabili aleatorie che stabiliscono la presenza o l'assenza di ciascun collegamento.

Uno dei modelli più studiati è il modello di Paul Erdős e Alfréd Rényi, noto come Erdős–Rényi model. In questo modello si considerano n vertici e ogni possibile arco tra due vertici distinti viene inserito indipendentemente con probabilità p .

Il grafo risultante rappresenta una realizzazione casuale tra tutti i grafi possibili con n vertici.

Capitolo 2

Catene di Markov

Questo capitolo è una trattazione delle catene di Markov e delle loro principali caratteristiche. Inizia definendo le loro matrici di transizione e la loro proprietà caratteristica: la proprietà di Markov.

Dopo aver introdotto il concetto di distribuzione stazionaria, si analizzano le nozioni di accessibilità, classi di comunicazione e periodo, che permettono di classificare il comportamento degli stati.

Successivamente si studiano proprietà come i tempi d'arresto e la proprietà di Markov forte, per poi introdurre i concetti di ricorrenza e transienza, legati alla classificazione dei diversi stati.

Si studia il comportamento asintotico della catena: la catena converge alla distribuzione stazionaria e perde memoria dello stato iniziale.

Si conclude con il teorema fondamentale di convergenza per le catene di Markov discrete.

Il contenuto di questo capitolo è basato principalmente su [2] e [3].

2.1 Catene di Markov e probabilità di transizione

2.1.1 Definizione

Definizione 11. Chiamiamo *catena di Markov a tempi discreti* una successione di variabili $X_0, X_1, \dots, X_n, \dots$ tali che per tutti gli stati i, j, k_0, k_1, \dots e per tutti i tempi $n=0, 1, 2, \dots$,

$$P(X_{n+1} = j \mid X_n = i, X_{n-1} = k_{n-1}, \dots) = p(i, j) \quad (2.1)$$

dove ciascuna X_n è una variabile aleatoria a valori in un insieme S , detto **spazio degli stati** (il quale deve essere finito o numerabile).

$p(i, j)$ dipende solo dagli stati i, j e non dal tempo n o dai precedenti stati k_{n-1}, k_{n-2}, \dots

I numeri $p(i,j)$ sono chiamati **probabilità di transizione della catena** (transizione dallo stato i allo stato j).

2.1.2 Esempio: random walk

Il **random walk** semplice sul reticolo degli interi Z^d è la catena di Markov le cui probabilità di transizione sono

$$p(x, x \pm e_i) = \frac{1}{2d} \quad \forall x \in Z^d \quad (2.2)$$

dove e_1, e_2, \dots, e_d sono i vettori unitari standard in Z^d ($Z^d =$ griglia infinita in d dimensioni).

Si sposta, ad ogni passo, verso uno dei vicini più prossimi scelto a caso.

Ogni mossa ha la stessa probabilità: $\frac{1}{2d}$

2.1.3 Proprietà di Markov

Proposizione 12. Se X_n è una catena di Markov con probabilità di transizione $p(x,y)$, allora per ogni sequenza di stati x_0, x_1, \dots, x_{m+n} ,

$$P(X_{m+i} = x_{m+i} \quad \forall 0 < i \leq n \mid X_i = x_i \quad \forall 0 < i \leq m) = \prod_{i=1}^n p(x_{m+i-1}, x_{m+i}) \quad (2.3)$$

Di conseguenza, le probabilità di transizione a n passi

$$p_n(x, y) := P(X_{n+m} = y \mid X_m = x) \quad (2.4)$$

dipendono solo dal ritardo temporale n e dagli stati iniziale e finale x, y , ma non dipendono da X_0, X_1, \dots, X_{m-1} .

In una catena di Markov, il futuro dipende solamente dallo stato attuale, e non dal passato.

$$P(X_{k+1} = y \mid X_0, \dots, X_k) = P(X_{k+1} = y \mid X_k) \quad (2.5)$$

Considerando una traiettoria specifica:

$$x_m \rightarrow x_{m+1} \rightarrow \dots \rightarrow x_{m+n} \quad (2.6)$$

la probabilità di seguirla è il prodotto delle probabilità di transizione ad ogni passo (grazie alla proprietà di Markov)

$$p(x_m, x_{m+1}) \cdot p(x_{m+1}, x_{m+2}) \cdot \dots \cdot p(x_{m+n-1}, x_{m+n}) \quad (2.7)$$

2.1.4 Matrice di transizione

Sia $\{X_n\}_{n \geq 0}$ a tempo discreto su uno spazio degli stati S , con probabilità di transizione $p(i, j)$.

I numeri $p(i, j)$ soddisfano le proprietà:

- $p(i, j) \in [0, 1]$
- $\sum_j p(i, j) = 1$ per ogni i

Queste $p(i, j)$ sono gli elementi della matrice che esprime le probabilità di transizione della catena; la riga i -esima è costituita dalle $p(i, j)$ per $j \in S$:

$$P = (p(i, j))_{i, j \in S} \quad (2.8)$$

Se S ha N elementi, allora si tratta di una matrice $N \times N$ (nel caso di S infinito, P è $\infty \times \infty$).

Per la legge della probabilità totale, la somma degli elementi di ogni riga è 1; Una matrice con questa proprietà si dice stocastica. Una **matrice stocastica** è anche una matrice quadrata non negativa (ha tutti gli elementi non negativi).

Si può invece definire **substocastica** quando è sempre quadrata non negativa, ma con la differenza che la somma di elementi su ogni riga è < 1 ; si dice invece **doppiamente stocastica** quando sia la somma su ogni riga che quella su ogni colonna è 1.

A volte è utile anche considerare la probabilità di transizione ad n passi:

$$p_n(i, j) = P(X_n = j \mid X_0 = i) \quad (2.9)$$

Usando il simbolismo matriciale, si può dire che:

$$p_n(i, j) = (P^n)_{ij} \quad (2.10)$$

2.1.5 Le equazioni di Chapman-Kolmogorov

Proposizione 13. *Le probabilità di transizione a n passi $p_n(i, j)$ sono gli elementi della n -esima potenza della matrice P (P^n) e soddisfano le **equazioni di Chapman-Kolmogorov**:*

$$p_{n+m}(i, j) = \sum_{k \in S} p_n(i, k) p_m(k, j) \quad (2.11)$$

(dove ricordiamo che $p_n(i, j)$ è la probabilità di passare dallo stato i allo stato j in n passi)

Per andare da i a j in $n + m$ passi il sistema:

1. parte da i

2. in n passi arriva in uno stato intermedio k (dato che non sappiamo quale, dobbiamo sommare le probabilità considerandoli tutti)
3. da k arriva a j in ulteriori m passi

Esiste anche una versione matriciale delle equazioni di Chapman-Kolmogorov:

$$P^{n+m} = P^n P^m \quad (2.12)$$

2.1.6 Evoluzione della distribuzione e stazionarietà

Supponiamo uno stato iniziale X_0 casuale, con distribuzione ν :

$$P_\nu\{X_0 = i\} = \nu(i) \quad (2.13)$$

per tutti gli stati $i \in S$.

Per le equazioni di Chapman-Kolmogorov e per la legge della probabilità totale,

$$P_\nu\{X_n = j\} = \sum_i \nu(i) p_n(i, j) \quad (2.14)$$

Equivalentemente, se la distribuzione iniziale è ν^T , dopo n passi la distribuzione sarà $\nu^T P^n$ (vediamo la distribuzione iniziale ν^T come un vettore riga).

Si può osservare che, se esiste una distribuzione di probabilità ν su S tale che

$$\nu^T P = \nu^T \quad (2.15)$$

allora, per ogni $n \geq 1$

$$\nu^T P^n = \nu^T \quad (2.16)$$

Se la catena ha distribuzione iniziale ν , la distribuzione marginale di X_n è ν per ogni $n \geq 1$; tale distribuzione si dice *stazionaria*.

Definizione 14. Una distribuzione di probabilità π su S è **stazionaria** se

$$\pi^T P = \pi^T \quad (2.17)$$

2.1.7 Accessibilità e classi di comunicazione

Definizione 15. Uno stato j si dice **accessibile** da uno stato i se esiste un percorso a probabilità positiva da i a j , cioè se esiste una sequenza finita di stati k_0, k_1, \dots, k_m tali che $k_0 = i$, $k_m = j$ e $p(k_t, k_{t+1}) > 0$ per ogni $t = 0, 1, \dots, m-1$. Due stati i e j comunicano se ciascuno è accessibile all'altro.

La comunicazione $i \leftrightarrow j$ è una relazione di equivalenza (quindi transitiva: se $i \leftrightarrow j$ e $j \leftrightarrow k$, allora $i \leftrightarrow k$).

Tutti gli stati che comunicano fra loro formano una **classe di comunicazione** (stati di classi diverse non comunicano fra loro).

2.1.8 Periodo

Definizione 16. Il **periodo** di uno stato i è il massimo comune divisore dell'insieme

$$\{n \in \mathbb{N} : p_n(i, i) > 0\} \quad (2.18)$$

Se ogni stato ha periodo 1, la catena di Markov si dice **aperiodica**; una catena aperiodica non ha cicli fissi.

Esiste un modo semplice per capire se la catena è aperiodica: se c'è almeno uno stato con probabilità di restare nello stesso stato in un passo, allora il periodo è 1.

Se i non è accessibile da se stesso, definiamo il periodo come $+\infty$.

Se due stati comunicano fra loro, devono avere lo stesso periodo. Di conseguenza, se la catena è **irriducibile** (tutti gli stati comunicano fra loro), allora tutti gli stati hanno lo stesso periodo.

Se la catena di Markov ha una distribuzione stazionaria π per cui $\pi(i) > 0$, e se gli stati i, j comunicano, allora $\pi(j) > 0$.

2.2 Catene di Markov a spazio di stati finito

Teorema 17. Una catena di Markov X_n irriducibile su uno spazio degli stati finito ha un'unica distribuzione stazionaria π .

Se la catena non è solo irriducibile ma anche aperiodica, avremo per qualsiasi distribuzione iniziale ν ,

$$\lim_{n \rightarrow \infty} P^\nu \{X_n = j\} = \pi(j) \quad \forall j \in S \quad (2.19)$$

Se lo spazio degli stati è finito e tutti gli stati comunicano (cioè la catena è irriducibile), allora nel lungo periodo la catena si stabilizzerà in uno stato di equilibrio (a prescindere dalla condizione iniziale).

2.3 Tempi di arresto, proprietà di Markov forte

2.3.1 Tempi di arresto

Definizione 18. Sia $\{X_n\}_{n \geq 0}$ una catena di Markov su uno spazio degli stati finito o numerabile.

Un **tempo di arresto** è una variabile casuale T che assume valori nell'insieme $\mathbb{N}^+ \cup \{\infty\}$ tale che, per ogni $m \in \mathbb{N}^+$, l'evento $\{T = m\}$ è determinato dai valori X_0, X_1, \dots, X_m .

Un tempo di arresto non è altro che una variabile aleatoria T tale che, per ogni istante m , la decisione di arrestare il processo al tempo m dipende esclusivamente dall'evoluzione del processo fino a quell'istante e non da valori futuri.

Un semplice esempio di tempo di arresto può essere: $T =$ il primo istante in cui la catena entra nello stato 0.

Per sapere se $T = m$ bisogna controllare se $X_m = 0$ e se $X_0, \dots, X_{m-1} \neq 0$.

Non è invece un esempio di tempo di arresto $T =$ l'ultimo istante in cui la catena visita lo stato 0 (presuppone conoscenza del futuro).

2.3.2 Proprietà di Markov forte

Proposizione 19. *Sia T un tempo di arresto per la catena di Markov $\{X_n\}_{n \geq 0}$. La catena di Markov si rigenera completamente al tempo T , ovvero il futuro X_{T+1}, X_{T+2}, \dots è indipendente dal passato X_0, X_1, \dots, X_{T-1} , dato il valore di T e lo stato $X_T = x$ al tempo T .*

Più precisamente, per ogni $m < \infty$ e per tutti gli stati $x_0, x_1, \dots, x_{m+n} \in S$ tali che l'evento $T = m$ sia possibile, vale

$$P(X_{T+i} = x_{m+i} \forall 1 \leq i \leq n \mid T = m \text{ and } X_i = x_i \forall 0 \leq i \leq m) = \prod_{i=1}^n p(x_{m+i-1}, x_{m+i}) \quad (2.20)$$

In sostanza la **proprietà di Markov forte** afferma che una catena di Markov, osservata ad un tempo di arresto, riparte come nuova.

2.4 Ricorrenza e transienza

2.4.1 Definizioni

Definizione 20. *Sia $\{X_n\}_{n \geq 0}$ una catena di Markov su uno spazio degli stati finito o numerabile S e, per ogni stato x , sia $T_x = T_x^{(1)}$ il tempo di primo ingresso nello stato x .*

Uno stato x si dice:

- **ricorrente** se $P_x\{T_x < \infty\} = 1$
- **transiente** se $P_x\{T_x < \infty\} < 1$
- **positivamente ricorrente** se $E_x[T_x] < \infty$
- **nullamente ricorrente** se è ricorrente ma $E_x[T_x] = \infty$

Esistono condizioni alternative per transienza e ricorrenza:

Supponiamo che lo stato x sia ricorrente: se la catena parte da $X_0 = x$, è certo che ritornerà in x . Se in base alla proprietà di Markov forte la catena si rigenera al tempo $T_x^{(1)}$ del primo ritorno, cioè si comporta come una nuova versione della catena che parte dallo stato x . Lo stato x verrà quindi visitato una seconda volta e, per induzione, almeno k volte, per ogni k . Uno stato ricorrente x verrà quindi visitato infinite volte.

$$E^x N_x = \sum_{n=0}^{\infty} p_n(x, x) = \infty \quad (2.21)$$

dove

$$N_x = \sum_{n=0}^{\infty} 1\{X_n = x\} \quad (2.22)$$

N_x indica il numero totale di volte in cui la catena visita lo stato x (1 se al tempo n sei in x , 0 altrimenti).

Mentre, per uno stato x transiente, non è più certo che x venga rivisitato, ma nel caso in cui questo avvenga, la catena si rigenera al tempo $T_x^{(1)}$, grazie alla proprietà di Markov forte. Per ogni $k = 1, 2, \dots$

$$P_x\{T_x^{(k)} < \infty\} = P_x\{T_x < \infty\}^k \quad (2.23)$$

Corollario 21. *Ricorrenza e transienza sono proprietà di classe; ad esempio, se x è ricorrente e x comunica con y , allora anche y è ricorrente.*

Inoltre, in una catena di Markov irriducibile, tutti gli stati sono dello stesso tipo (ricorrenti o transienti). Chiameremo la catena ricorrente o transiente (come per positivamente o nullamente ricorrente).

2.4.2 Teorema di Polya

Teorema 22. *Il random walk semplice in Z^d è:*

- *ricorrente se $d = 1, 2$*
- *transiente se $d \geq 3$*

2.5 Catena delle escursioni

Sia X_n una catena di Markov irriducibile ricorrente su uno spazio degli stati finito o numerabile, con matrice delle probabilità di transizione P . Supponiamo anche che la catena parta dallo stato $X_0 = x$.

Possiamo tenere una lista casuale degli stati visitati, usando la regola:

- elemento iniziale della lista: x
- per ogni $n = 1, 2, \dots$ aggiungere lo stato X_n alla fine della lista se $X_n \neq x$
- se invece $X_n = x$, cancellare tutto il contenuto della lista tranne l'elemento x

La sequenza di liste casuali così prodotta è chiamata *catena delle escursioni*.

Definizione 23. *La catena delle escursioni da x è la catena di Markov sullo spazio degli stati W , con probabilità di transizione definite da:*

- $q(xy_1y_2\dots y_k, xy_1y_2\dots y_ky_{k+1}) = p(y_k, y_{k+1})$ se $y_{k+1} \neq x$
- $q(xy_1y_2\dots y_k, x) = p(y_k, x)$
- $q(x, xy) = p(x, y)$ se $y \neq x$
- $q(w, w') = 0$ in tutti gli altri casi

2.5.1 Esempio di catena delle escursioni

Se la sequenza degli stati visitati dalla catena X_n è $x, y_1, y_2, x, y_3, y_4, \dots$, allora gli stati successivi della catena delle escursioni sono $x, xy_1, xy_1y_2, x, xy_3, xy_3y_4, \dots$

Le liste che possono essere stati della catena delle escursioni sono le parole finite $xy_1y_2\dots y_k$ di lunghezza ≥ 1 tali che:

1. x compare una volta sola nella parola, all'inizio
2. per ogni coppia di lettere adiacenti, la probabilità di transizione è > 0 nella catena X_n

Indichiamo con W l'insieme delle parole.

2.5.2 Proiezione sull'ultima lettera

Sia $F : W \rightarrow S$ la proiezione sull'ultima lettera, ovvero l'applicazione che associa ad ogni parola $xy_1\dots y_k$ la sua ultima lettera y_k .

Lemma 24. *Sia Y_n una versione della catena delle escursioni, cioè una catena di Markov sullo spazio degli stati W con matrice di transizione $Q = (q(u, v))_{u, v \in W}$. Allora $F(Y_n)$ è una versione della catena originale X_n ; equivalentemente, $F(Y_n)$ è una catena su S con la stessa matrice di transizione della catena originale.*

2.5.3 Distribuzione stazionaria della catena delle escursioni

Supponiamo che la catena delle escursioni abbia una distribuzione stazionaria ν , che per definizione deve soddisfare il sistema di equazioni

$$\nu^T = \nu^T Q \quad (2.24)$$

dove Q è la matrice di transizione della catena delle escursioni.

Se $w \in W$ è una parola di lunghezza ≥ 2 , esiste una sola parola w' tale che $q(w', w) > 0$, ovvero la parola ottenuta cancellando l'ultima lettera di w .

L'equazione di stato stazionario per $\nu(w)$ diventa:

$$\nu(w) = \nu(w')q(w', w) \quad (2.25)$$

Iterando fino alla lettera x otteniamo:

$$\nu(xy_1 \dots y_k) = \nu(x)p(x, y_1) \prod_{i=1}^{k-1} p(y_i, y_{i+1}) \quad (2.26)$$

Da questa formula segue che può esistere al massimo una distribuzione stazionaria per la catena delle escursioni.

Inoltre, una distribuzione stazionaria esiste se e solo se esiste un valore finito e positivo $\nu(x)$ tale che:

$$\sum_{k=0}^{\infty} \sum_{y_1, \dots, y_k} \nu(x)p(x, y_1) \prod_{i=1}^{k-1} p(y_i, y_{i+1}) = 1 \quad (2.27)$$

Proposizione 25. *La catena delle escursioni ha una distribuzione di probabilità stazionaria ν se e solo se x è uno stato positivamente ricorrente della catena X_n , cioè se $E_x[T_x] < \infty$ (ritorna in x in un tempo medio finito).*

In questo caso, la distribuzione stazionaria è data da

$$\nu(x) = \frac{1}{E_x[T_x]} \quad (2.28)$$

Corollario 26. *Segue che se una catena di Markov irriducibile ha uno stato positivamente ricorrente x , allora essa possiede una distribuzione stazionaria π per cui:*

$$\pi(x) = \frac{1}{E_x[T_x]} \quad (2.29)$$

2.6 Escursioni e legge forte dei grandi numeri

In questa sezione considereremo solo escursioni complete, cioè segmenti della catena di Markov compresi tra le visite successive a uno stato distinto x .

Sia X_n catena irriducibile e ricorrente su uno spazio degli stati finito o numerabile, con matrice di transizione P .

Fissiamo x e per semplicità poniamo $\tau(k) = T_x^{(k)}$ per $k = 1, 2, \dots$

I tempi $\tau(k)$ indicano le visite successive allo stato x . Poniamo anche $\tau(0) = 0$.

Le escursioni dallo stato x sono le seguenti parole:

$$W_1 := (X_0, X_1, X_2, \dots, X_{\tau(1)-1}) \quad (2.30)$$

$$W_2 := (X_{\tau(1)}, X_{\tau(1)+1}, X_{\tau(1)+2}, \dots, X_{\tau(2)-1}) \quad (2.31)$$

$$\vdots \quad (2.32)$$

Poichè la catena è ricorrente, i tempi di arresto $\tau(k)$ sono tutti finiti e tutte le escursioni teminano.

Lemma 27. *In base a da dove parte la catena le escursioni hanno caratteristiche diverse:*

- sotto P^x , le escursioni W_1, W_2, \dots sono indipendenti e identicamente distribuite.
- sotto P^y (dove $y \neq x$), le escursioni W_1, W_2, \dots sono indipendenti, e W_2, W_3, \dots sono identicamente distribuite.

Sotto P^x sono indipendenti perchè ogni nuova escursione riparte da x come se fosse la prima volta, e identicamente distribuite perchè hanno sempre la stessa legge probabilistica, partendo sempre dallo stesso stato. Se invece parte da $y \neq x$, la prima escursione W_1 è diversa dalle altre.

Corollario 28. *Se esiste uno stato positivamente ricorrente x , allora tutti gli stati sono positivamente ricorrenti.*

2.6.1 Definizione: estensione additiva

Definizione 29. *Sia $f : S \rightarrow \mathcal{R}$ una funzione a valori reali definita sullo spazio degli stati S .*

L'estensione additiva di f all'insieme delle parole finite con lettere in \mathcal{X} è la funzione f_+ che assegna ad una parola finita $w = (x_1, x_2, \dots, x_m)$ il valore

$$f_+(w) := \sum_{i=1}^m f(x_i) \quad (2.33)$$

(Non fa altro che contare il numero di passi da cui è composta l'escursione).

2.6.2 Esempio: catena con 3 stati

Supponiamo che la catena abbia 3 stati: $X = \{x, a, b\}$.

Definiamo f così:

- $f(x) = 2$ (se sei nello stato x , il valore di f è 2)
- $f(a) = 1$
- $f(b) = 3$

La somma di f lungo un percorso si chiama f_+ .

Se ad esempio il percorso è (x, a, b) ,

$$f_+(x, a, b) = f(x) + f(a) + f(b) = 2 + 1 + 3 = 6 \quad (2.34)$$

2.6.3 Equilibrio asintotico

Corollario 30. *Sia $f : S \rightarrow \mathcal{R}$ una funzione non negativa (o limitata), e sia f_+ la sua estensione additiva.*

Per ogni stato iniziale $y \in S$, con probabilità 1 sotto P^y ,

$$\lim_{k \rightarrow \infty} \frac{1}{k} \sum_{i=1}^k f_+(W_i) = E^x[f_+(W_1)] = E^x \sum_{j=0}^{\tau(i)-1} f(X_j) \quad (2.35)$$

Stiamo sostanzialmente facendo una media su tutte le escursioni, la quale converge ad un valore fisso.

In particolare (usando $f = 1$) con probabilità 1 sotto P^y ,

$$\lim_{k \rightarrow \infty} \frac{\tau(k)}{k} = E^x[\tau(1)] \quad (2.36)$$

Dove $\tau(k)$ è il tempo del k -esimo ritorno a x e $E^x[\tau(1)]$ la durata media dell'escursione.

In questo caso, dove ogni passo vale 1, possiamo dire che la durata media di un'escursione converge alla durata media teorica.

In altre parole, quando il numero di ritorni k diventa grande, il tempo del k -esimo ritorno cresce come $k \cdot$ durata media di un'escursione.

Chiamiamo N_n^x il numero di visite a x fino al tempo n .

Ora invertiamo la relazione $\tau(k) \approx kE^x[\tau(1)]$:

$$n \approx N_n^x \cdot E^x[\tau(1)] \quad (2.37)$$

Da cui segue:

$$N_n^x \approx \frac{n}{E^x[\tau(1)]} \quad (2.38)$$

In altre parole possiamo dire che il numero di visite a x cresce linearmente con il tempo, con pendenza $\frac{1}{E^x[\tau(1)]}$.

Se la catena è positivamente ricorrente, esiste una distribuzione stazionaria π con

$$\pi(x) = \frac{1}{E^x[\tau(1)]} \quad (2.39)$$

La frazione di tempo trascorsa in x è esattamente $\pi(x)$.

Inoltre, se la catena è nullamente ricorrente avremo che $E^x[\tau(1)] = \infty$, perciò la frazione di tempo trascorsa in x tenderà a 0: passi infinite volte da x , ma ogni stato riceve una frazione di tempo nulla nel lungo periodo.

2.6.4 Teorema di sintesi

Teorema 31. *positivamente ricorrente e irriducibile:*

$$\lim_{n \rightarrow \infty} \frac{N_n^x}{n} = \pi(x) \quad (2.40)$$

nullamente ricorrente e irriducibile:

$$\lim_{n \rightarrow \infty} \frac{N_n^x}{n} = 0 \quad (2.41)$$

La catena visita ogni stato infinite volte, poichè irriducibile; la differenza è solo il tempo in cui resta in ciascuno stato.

2.7 Accoppiamento e Teorema del limite di Kolmogorov

Teorema 32. *Supponiamo che X_n sia una catena di Markov aperiodica, positivamente ricorrente e irriducibile su uno spazio degli stati S , e sia π la sua unica distribuzione stazionaria.*

*Allora, per ogni coppia di stati x e y abbiamo il **teorema fondamentale di convergenza per le catene di Markov discrete**:*

$$\lim_{n \rightarrow \infty} P^x\{X_n = y\} = \pi(y) \quad (2.42)$$

Dove $P^x\{X_n = y\}$ è la probabilità di essere nello stato y al tempo n partendo da x :
Quando il tempo n diventa grande, questa probabilità non dipende più da x e converge a $\pi(y)$, dove $\pi(y)$ descrive la percentuale di tempo che la catena passa nello stato y , nel lungo periodo.

Questo teorema formalizza l'idea che una catena di Markov "si dimentica" dello stato iniziale e converge a un equilibrio unico.

Capitolo 3

Simulazione in python di passeggiate aleatorie dentro labirinti

Questo capitolo contiene uno studio sperimentale di alcune delle caratteristiche delle catene di Markov; in particolare, si concentra sul random walk.

Viene usata una simulazione Python che genera labirinti, dentro cui si muove il random walker.

I risultati presentati in questa tesi sono stati ottenuti tramite simulazioni Monte Carlo, in cui il comportamento del walker è stato campionato in maniera casuale con un grande numero di repliche.

Esiste anche un'interessante teoria matematica che approfondisce questo argomento, ma la sua trattazione esula dagli scopi della presente tesi.

Questo capitolo è tratto prevalentemente da [4], [5], [6], [7].

3.1 Catene di Markov nei labirinti

Il labirinto costituisce un modello intuitivo di catena di Markov: possiamo interpretare ogni cella come uno stato della catena, e i passaggi fra celle adiacenti come possibili transizioni fra stati.

Rispetta la proprietà di Markov: le transizioni fra stati discreti dipendono solo dallo stato attuale, e non dai precedenti.

Il movimento nel labirinto segue il modello matematico del random walk (passeggiata aleatoria, sezione 2.1.2), in cui ogni passo viene scelto casualmente tra le direzioni possibili.

La differenza principale rispetto al random walk classico è la presenza dei muri, che limitano le direzioni disponibili.

Il labirinto è composto da muri perimetrali, che restano fissi ad eccezione dell'uscita, e da muri interni, la cui presenza può variare secondo diverse probabilità. In questo senso la struttura del labirinto può essere interpretata come un grafo aleatorio.

Considerando ogni cella del labirinto come un vertice del grafo e i passaggi tra celle confinanti come archi, la distribuzione casuale dei muri ne determina la struttura. In questo modo, si ottengono diverse configurazioni dello spazio degli stati su cui studiare il comportamento della catena di Markov e della camminata aleatoria tramite simulazioni Monte Carlo.

3.1.1 Classificazione degli stati

L'uscita può essere classificata come stato assorbente, poichè determina la fine del percorso nel labirinto, dove non sono possibili ulteriori transizioni. Se lo stato i è assorbente, $P(i \rightarrow i) = 1$.

Uno stato assorbente è un caso particolare degli stati ricorrenti: lo stato viene visitato e in più il walker si ferma.

Tutte le altre celle accessibili del labirinto sono stati transienti. [4] [7]

Se viene invece chiusa l'uscita, tutti gli stati accessibili diventano ricorrenti, poichè il walker visita ognuno infinite volte (non ci sono più stati transienti).

Un'importante conseguenza pratica è l'irriducibilità della catena: ogni stato accessibile diventa raggiungibile da ogni altro stato.

Inoltre, la catena è solitamente aperiodica (ad esclusione di labirinti particolarmente regolari o ciclici); esiste perciò una distribuzione stazionaria unica su tutte le celle accessibili (sezione 2.3).

3.1.2 Struttura

Il labirinto è costituito da una griglia quadrata $d \times d$, le cui celle rappresentano gli stati discreti della catena.

Partendo dalla cella in alto a sinistra, l'entrata, è possibile muoversi di solo un passo alla volta verso uno degli stati adiacenti (verso destra, sinistra, l'alto o il basso); le mosse consentite dipendono dalla struttura del labirinto.

Le probabilità di transizione (p_{ij}) verso le diverse direzioni sono:

- $\frac{1}{d(i,j)}$ per le celle adiacenti e raggiungibili (sezione 2.1.2)
($d(i,j)$ è il numero di celle accessibili a i,j)
- 0 per le celle non accessibili

Le mosse consentite sono equiprobabili.

3.2 Generazione del labirinto

Usando python viene creata una griglia alternata celle/muri, dove inizialmente le celle sono tutte isolate.

```
def create_maze(h,w): #crea il labirinto
    maze = []
    for row in range(h):
        maze_row=[]
        for col in range(w):
            if row % 2 == 1 and col % 2 == 1:
                maze_row.append(cell) #cella
            elif row % 2 == 0 and col % 2 == 0:
                maze_row.append(intersection) #incrocio
            else:
                maze_row.append(wall) #muro
        maze.append(maze_row)
    return maze
```

```
dim = 10
R = dim #numero di righe
C = dim #colonne
height = 2*R + 1
width = 2*C + 1
start = (0, 0) #alto a sx, coordinata del labirinto
end = (R-1, C-1) #basso a dx
```

height e *width* corrispondono al numero di righe e colonne della griglia con muri e celle, mentre $d \times d$ è il numero delle sole celle.

I muri interni vengono aperti con una probabilità $p \in [0, 1]$, la quale determina la connettività del labirinto. Ad esempio, per $p = 0.5$ vengono rimossi metà dei muri, per $p = 1$ tutti.

Quelli esterni rimangono invece sempre tutti chiusi.

```
def open_random_walls(maze, height, width, p): #p deve essere
    compresa fra 0 e 1
    for r in range(1, height-1):
        for c in range(1, width-1):
            if maze[r][c] == wall:
                if random.random() < p:
                    maze[r][c] = ex_wall
```

Per garantire la non banalità della simulazione, vengono usati solamente labirinti dove è garantito almeno un percorso possibile fra entrata $(0, 0)$ ed uscita $(R - 1, C - 1)$.

```

def exists_path(maze, R, C, max_steps=100000, reflection_p = 0):
    i,j=0,0 #cella iniziale
    for _ in range(max_steps):
        if (i, j) == (R-1, C-1):
            return True
        i, j = random_direction(i, j, maze, R, C, reflection_p)
    return False

```

3.3 Obiettivi

La simulazione di camminata aleatoria su labirinti casuali costituisce uno strumento molto utile per osservare molte caratteristiche e comportamenti delle catene di Markov.

Ad esempio, attraverso la simulazione numerica è possibile stimare la probabilità che il walker raggiunga l'uscita (cella particolare vista come stato assorbente) entro un numero di passi fissato a priori.

Insieme a questa probabilità, nei soli casi in cui il walker raggiunga l'uscita, può essere interessante vedere il numero di passi necessari per arrivarci.

Uno degli obiettivi principali della simulazione è indagare come il variare della dimensione del labirinto d influenzi la probabilità che il walker raggiunga l'uscita e il numero di passi necessari per raggiungerla.

Indicando con p la probabilità di apertura di un collegamento tra celle adiacenti, è possibile studiare anche la relazione fra tali grandezze e il grado di connettività del labirinto, confrontando i risultati ottenuti per diversi valori di p .

Interessante è il caso con il parametro $p = 1$ (tutti i muri aperti), poichè corrisponde al random walk classico $2D$ su griglia.

La simulazione viene complicata ulteriormente sostituendo alcuni dei muri con degli specchi, modifica attuata secondo una certa probabilità. Si può indagare in quale modo questa probabilità influenzi la dinamica della camminata aleatoria, in particolare la probabilità che il walker raggiunga l'uscita e il numero di passi necessari ad uscire.

L'ultimo obiettivo riguarda la proprietà di Markov, elemento centrale nella teoria delle catene di Markov. In particolare, si analizzano le conseguenze della sua violazione, valutando come una dinamica non markoviana influenzi il comportamento del walker.

A tal fine si aggiunge un'importante regola alla camminata aleatoria: ad ogni passo, il walker può dirigersi in tutte le celle adiacenti, ad eccezione dell'ultima cella visitata prima di quella corrente (le altre celle rimangono equiprobabili).

In questo modo si aggiunge alla dinamica una breve memoria, modificando le probabilità di transizione della catena.

3.4 Simulazione random walk nel labirinto

La simulazione usa una funzione che, partendo dalla cella in cui si trova il walker, seleziona una direzione casuale fra quelle disponibili (celle adiacenti non separate da muri).

```
def random_direction(i, j, maze, R, C, reflection_p): #random
walk
moves = []
# su
if i > 0 and maze[2*i][2*j+1] == ex_wall:
    moves.append((-1, 0))
# giu
if i < R-1 and maze[2*i+2][2*j+1] == ex_wall:
    moves.append((1, 0))
# sinistra
if j > 0 and maze[2*i+1][2*j] == ex_wall:
    moves.append((0, -1))
# destra
if j < C-1 and maze[2*i+1][2*j+2] == ex_wall:
    moves.append((0, 1))
if not moves:
    return i, j
di, dj = random.choice(moves)
di, dj = mirror(di, dj, reflection_p)
ni, nj = i + di, j + dj
if 0 <= ni < R and 0 <= nj < C:
    return ni, nj
return i, j
```

Si genera un percorso casuale che parte sempre dall'entrata (cella in alto a sinistra) e si sviluppa all'interno del labirinto.

```
def path(maze, R, C, start, end, reflection_p):
path_list=[start]
steps = 0 #contatore dei passi
while True:
    i,j=path_list[-1]
    next_state = random_direction(i, j, maze, R, C,
    reflection_p)
    path_list.append(next_state)
    steps += 1
    if next_state == end or steps >= 100:
        break
return (steps, path_list)
```

3.5 Rappresentazione grafica del labirinto e tracciamento del percorso

Trovato un percorso, è possibile tracciarlo graficamente.

```
def draw_maze_with_path(maze, path_list):
    path_cells = {(2*i+1, 2*j+1) for (i, j) in path_list} #agiamo
        sulla griglia con i muri

    for r in range(len(maze)):
        line = ""
        for c in range(len(maze[0])):
            if r == 2*start[0] + 1 and c == 2*start[1] + 1:
                line += "ST"
            elif r == 2*end[0] + 1 and c == 2*end[1] + 1:
                line += "EN"
            elif (r, c) in path_cells:
                line += "++"
            elif maze[r][c] == wall:
                line += " " #quadretto colorato
            else:
                line += " " #quadretto bianco
        print(line)
```

I quadrati colorati indicano i muri, quelli bianchi le celle. "ST" ed "EN" rappresentano rispettivamente entrata ed uscita, mentre il percorso è segnalato da "++".



Figura 3.1: Labirinto 15×15 , muri aperti con probabilità 0.5

3.6 Probabilità di uscita e numero medio di passi

La simulazione si avvale di una funzione che ha lo scopo di stimare la probabilità che il random walker esca dal labirinto entro un certo numero di passi fissato.

Questa funzione genera casualmente n cammini diversi sullo stesso labirinto e registra quanti di questi arrivano alla fine.

```
def exit_probability(maze, R, C, start, end, n,
steps_n, reflection_p): #probabilita che il walker esca dal
labirinto entro un certo numero di passi
    success = 0
    for _ in range(n): #numero di cammini diversi
        i, j = start
        for _ in range(steps_n):
            if (i, j) == end:
                success += 1
                break
            i, j = random_direction(i, j, maze, R, C,
reflection_p)
    return success / n
```

Viene inoltre implementata una funzione di stima del tempo medio di uscita, espresso in termini di numero di passi compiuti dal walker per raggiungere l'uscita del labirinto.

Anche questa genera n cammini diversi, ma utilizza e va a mediare solo quelli arrivati con successo all'uscita.

Per ogni cammino viene fissato un numero massimo volutamente alto di passi disponibili per arrivare all'uscita, in modo da evitare loop infiniti.

```
def average_steps(maze, R, C, start, end, n, max_steps,
                 reflection_p):
    steps_list = [] #solo cammini che escono
    for _ in range(n):
        i, j = start
        steps = 0
        while steps < max_steps and (i, j) != end:
            i, j = random_direction(i, j, maze, R, C,
                                   reflection_p)
            steps += 1
        if (i, j) == end:
            steps_list.append(steps)
    if len(steps_list) == 0:
        return np.nan #valore indefinito
    return np.mean(steps_list)
```

In tutta la simulazione, ad ogni richiamo delle funzioni vengono prodotti $n = 100$ cammini diversi.

3.7 Distribuzione stazionaria e varianza

Come precedentemente introdotto nella sezione 2.1.6, una distribuzione di probabilità π (cioè tale che $\sum \pi_i = 1, \pi_i \in [0, 1]$ per ogni i) si dice stazionaria per la catena di Markov descritta dalla matrice di transizione P se $\pi = \pi P$.

Per trovare la distribuzione stazionaria all'interno del labirinto simulato si può pensare di chiudere l'uscita, lasciando evolvere il random walk per un numero sufficientemente grande fissato di passi all'interno del labirinto.

Il sistema converge alla distribuzione stazionaria, definita come la frequenza di visita delle singole celle da parte del walker, normalizzata ad una distribuzione di probabilità.

```
def stationary_distribution(maze, R, C, start, stationary_steps,
                          reflection_p):
    counts = np.zeros((R, C)) #crea matrice di zeri
    i, j = start
    for _ in range(stationary_steps): #numero di passi per
        #trovare la distribuzione stazionaria
```

```

    counts[i, j] += 1
    i, j = random_direction(i, j, maze, R, C, reflection_p)
    return counts / counts.sum() #normalizza la probabilita

```

Si può anche stimare la varianza della distribuzione stazionaria, che quantifica quanto le probabilità legate alle diverse celle differiscono fra loro.

```

def stationary_variance(maze, R, C, start, stationary_steps,
    reflection_p):
    a = stationary_distribution(maze, R, C, start,
        stationary_steps, reflection_p)
    return np.var(a)

```

Nei casi con una varianza piccola, le celle hanno probabilità simili e il labirinto è abbastanza regolare, avvicinandosi di più ad un random walk classico.

Una varianza elevata è invece indice della disomogeneità della frequenza di visita fra le celle del labirinto: alcune zone vengono frequentate molte più di altre, e si trovano più facilmente vicoli ciechi, colli di bottiglia o corridoi forzati, che vincolano la diffusione del walker.

3.8 Confronto fra diverse configurazioni di labirinto

Al fine di confrontare labirinti diversi si può, per semplicità, restringere solo a quelli in cui è possibile almeno un percorso valido fra entrata ed uscita.

```

def new_maze(height, width, R, C, open_walls_prob, reflection_p):
    found = False
    while found == False:
        maze = create_maze(height, width) #creo nuovo labirinto
        open_random_walls(maze, height, width, open_walls_prob) #
            apre muri casuali
        if exists_path(maze, R, C, max_steps=10000, reflection_p
            = 0) == True:
            found = True
    return maze #ritorna un labirinto con almeno un percorso
        valido

```

3.8.1 Analisi al variare della dimensione

Si studia come la probabilità di uscita dal labirinto e il valore atteso del numero di passi necessari per l'uscita dipendano dalla dimensione del labirinto.

A questo fine, si considerano diverse dimensioni, indicate con d , e per ognuna di essere si generano 50 labirinti indipendenti. Tutti gli altri parametri sono invece fissati.

Per ciascuna dimensione vengono misurate la media campionaria e la deviazione standard, sia della probabilità di uscita sia del numero di passi necessari per l'uscita.

Le d usate per l'analisi sono:

```
dimensions = [2, 3, 5, 7, 10, 15, 20]
```

Si possono graficare media campionaria e deviazione standard, in funzione della dimensione d .

3.8.2 Analisi al variare della densità e confronto con random walk classico 2D

Tenendo invece fissa la dimensione, si analizza l'effetto della variazione del parametro p , che regola la probabilità di apertura dei muri interni.

La simulazione segue la struttura di quella precedente, e ha il fine di studiare come una differenza nella struttura interna del labirinto può modificare la probabilità che il walker possa uscire (sempre entro un certo limite di passi) e il numero medio di passi con cui lo fa.

Come nel caso precedente, per ogni p si misurano media campionaria e deviazione standard, sia della probabilità di uscita sia del numero di passi necessari per l'uscita, dalle quali si ricavano dei grafici.

Le p usate per l'analisi sono:

```
open_walls_probabilities = [0.4, 0.45, 0.5, 0.55, 0.6, 0.65,
                             0.7, 0.75, 0.8, 0.85, 0.9, 0.95, 1]
```

Quest'ultimo, con $p = 1$, è un caso limite particolarmente interessante: essendo completamente privo di muri interni, fornisce un ottimo esempio di random walk classico su griglia 2D, confinato in uno spazio dxd .

Questo caso permette di evidenziare bene in quale modo la presenza dei muri interni influenzi la dinamica del random walk, a parità di dimensione della griglia.

Per il confronto si usa $p = 0.5$.

Per ognuna delle due probabilità, il codice produce una heatmap che mette in evidenza la distribuzione stazionaria a cui converge la catena.

Per ottenere la distribuzione stazionaria, il walker percorre 1000000 di passi.

```
open_walls_probabilities_heatmap = [0.5, 1]
```

```
def heatmap(maze, stationary_steps, reflection_p):#creiamo una
heatmap
a = stationary_distribution(maze, R, C, start,
stationary_steps, reflection_p)
plt.figure()
```

```
plt.imshow(a, cmap='hot', interpolation='nearest', origin='
    upper', vmin=0, vmax=np.max(a))
plt.colorbar(label="Probabilità stazionaria")
plt.title("Heatmap distribuzione stazionaria")
plt.show(block=False)
```

3.8.3 Specchi riflettenti

La simulazione viene resa più complessa tramite l'aggiunta di specchi che riflettono orizzontalmente la mossa del walker con una certa probabilità.

Questi specchi non sono vincolati ai muri, la riflessione del walker avviene invece in modo probabilistico a ogni passo orizzontale.

Gli specchi sono stati ispirati dall'articolo [4].

Le probabilità di riflessione scelte per l'analisi sono:

```
reflection_probabilities = [0.0, 0.1, 0.2, 0.3, 0.4, 0.5,
    0.6, 0.7, 0.8, 0.9, 1]
```

```
def mirror(di, dj, reflection_p):
    if random.random() < reflection_p:
        return di, -dj #specchio
    return di, dj
```

La funzione mirror viene poi integrata dentro a quella che seleziona una direzione casuale per ogni passo della passeggiata aleatoria.

```
di, dj = mirror(di, dj, reflection_p)
```

Come nei casi precedenti, si creano 50 labirinti per ogni probabilità di riflessione e si misurano media campionaria e deviazione standard, sia della probabilità di uscita sia del numero di passi necessari per l'uscita, per poi ricavare dei grafici.

3.9 Cosa accadrebbe se venisse violata la proprietà di Markov?

A parità di struttura e parametri del labirinto, vengono confrontate una passeggiata aleatoria standard, che soddisfa la proprietà di Markov, e una passeggiata aleatoria con memoria, nella quale le probabilità di transizione dipendono anche dalla storia del processo, in particolare dall'ultima posizione occupata.

Questo confronto ci permette di comprendere in che modo la presenza di una breve memoria possa modificare la dinamica della passeggiata.

A tal fine sono create nuove funzioni che escludono l'ultima cella visitata dal walker da quelle accessibili.

```
def random_direction_informed(i, j, last_i, last_j, maze, R, C):
    for _ in range(10):
        ni, nj = random_direction(i, j, maze, R, C, 0)
        if (ni, nj) != (last_i, last_j): #se torno indietro,
            riprovo
            return ni, nj
    return ni, nj #se sono bloccato, torno indietro

def exists_path_informed(maze, R, C, max_steps=10000):
    i,j=0,0
    last_i, last_j = i, j #all'inizio non ho una cella
    precedente
    for _ in range(max_steps):
        if (i, j) == (R-1, C-1):
            return True
        ni, nj = random_direction_informed (i, j, last_i, last_j,
            maze, R, C)
        last_i, last_j = i, j #salvo da dove vengo
        i, j = ni, nj #mi muovo
    return False

def new_maze_informed(height, width, R, C, open_walls_prob):
    found = False
    while found == False:
        maze = create_maze(height, width) #creo nuovo labirinto
        open_random_walls(maze, height, width, open_walls_prob) #
            apre muri casuali
        if exists_path_informed(maze, R, C, max_steps=10000) ==
            True:
            found = True
    return maze #ritorna un labirinto con almeno un percorso
    valido

def exit_probability_informed(maze, R, C, start, end, n, steps_n)
: #probabilita che il walker esca dal labirinto entro un certo
numero di passi
success = 0
for _ in range(n): #numero di cammini diversi
    i, j = start
    last_i, last_j = i, j
    for _ in range(steps_n):
```

```

        if (i, j) == end:
            success += 1
            break
        ni, nj = random_direction_informed(i, j, last_i,
            last_j, maze, R, C)
        last_i, last_j = i, j
        i, j = ni, nj
    return success / n

def average_steps_informed(maze, R, C, start, end, n, max_steps):
    steps_list = [] #solo cammini che escono
    for _ in range(n):
        i, j = start
        last_i, last_j = i, j
        steps = 0
        while steps < max_steps and (i, j) != end:
            ni, nj = random_direction_informed(i, j, last_i,
                last_j, maze, R, C)
            last_i, last_j = i, j
            i, j = ni, nj
            steps += 1
        if (i, j) == end:
            steps_list.append(steps)
    if len(steps_list) == 0:
        return np.nan #valore indefinito
    return np.mean(steps_list)

```

Anche qua si creano 50 labirinti per ognuna delle due configurazioni, tutti con i medesimi parametri.

Si stimano media campionaria e deviazione standard, sia della probabilità di uscita, sia del numero di passi necessari per l'uscita.

Capitolo 4

Risultati della simulazione e conclusione

4.1 Analisi al variare della dimensione

4.1.1 Dati numerici e grafici

Le tabelle seguenti contengono i dati numerici prodotti dalla simulazione python relativi alla probabilità di uscita dal labirinto entro 1000 passi e al numero di passi necessari per uscire (media e deviazione standard) in funzione della dimensione del labirinto.

dimensione	media probabilità di uscita	deviazione standard	probabilità di uscita
2	1.0		0.0
3	1.0		0.0
5	0.99		0.01
7	0.94		0.07
10	0.68		0.13
15	0.26		0.09
20	0.09		0.04

dimensione	media numero di passi	deviazione standard numero di passi
2	4.9	1.7
3	26	11
5	150	65
7	375	139
10	836	337
15	2499	586
20	3773	405

Media della probabilità di uscita in funzione della dimensione del labirinto

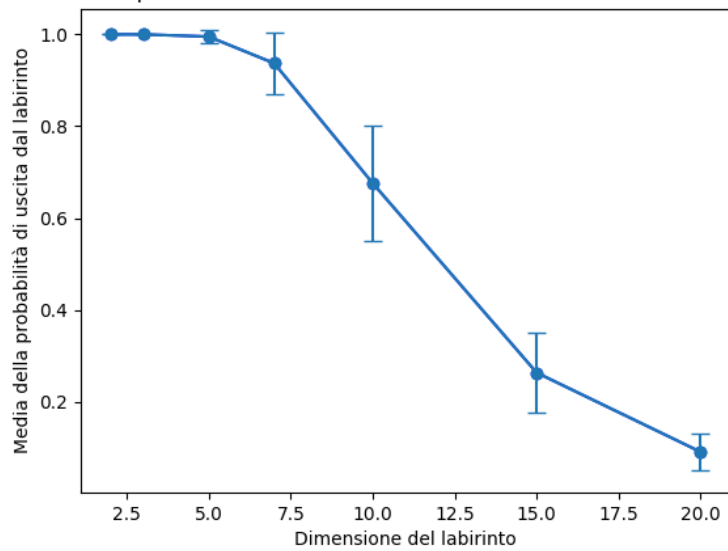


Figura 4.1: Media della probabilità di uscita da un labirinto entro 1000 passi in funzione della dimensione, $p = 0.7$

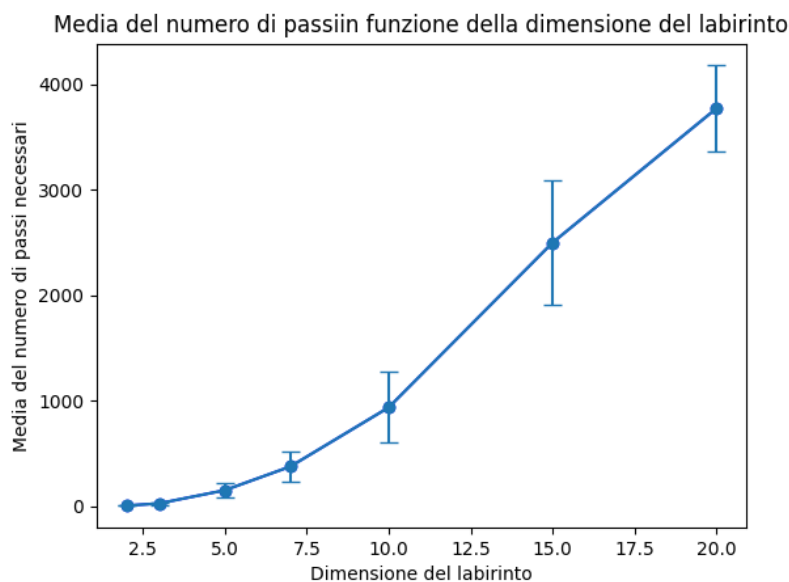


Figura 4.2: Media del numero di passi necessari ad uscire dal labirinto in funzione della dimensione, $p = 0.7$

4.1.2 Discussione dei risultati

Il primo grafico mostra un calo della probabilità di uscita entro 1000 passi all'aumentare della dimensione del labirinto.

Per dimensioni molto piccole del labirinto ($p = 2, 3$) la probabilità media di uscita è pari a 1 e la deviazione standard è nulla. Questo indica che in tutte le simulazioni il walker riesce sempre a uscire entro 1000 passi.

Aumentando il numero di celle, diventa più difficile raggiungere l'uscita, ed inizialmente aumenta la deviazione standard, segno di una maggiore variabilità del campione.

Per labirinti ancora più grandi ($p = 15, 20$) la probabilità media diventa molto bassa. In questo caso la deviazione standard diminuisce leggermente perché quasi tutte le simulazioni portano allo stesso risultato: il walker non riesce a uscire entro 1000 passi.

Il secondo grafico mostra che invece il numero di passi necessari per uscire aumenta notevolmente all'aumentare del numero di celle. Ad esempio, servono mediamente 4.9 passi per uscire dal labirinto 2×2 , e 3773 per uscire da quello 20×20 .

Anche qua, la deviazione standard segue lo schema dell'altro grafico: bassa nella parte iniziale e quella finale, elevata nella parte centrale, dove c'è una maggiore variabilità del campione. Negli estremi della curva, i comportamenti sono molto più stabili.

4.2 Analisi al variare della densità

4.2.1 Dati numerici e grafici

Le tabelle seguenti contengono i dati numerici prodotti dalla simulazione python relativi alla probabilità di uscita dal labirinto 10×10 entro 1000 passi e al numero di passi necessari per uscire (media e deviazione standard) in funzione della connessione (probabilità di apertura dei muri).

connessione	media probabilità di uscita	deviazione standard	probabilità di uscita
0.4	0.63		0.15
0.5	0.55		0.15
0.6	0.53		0.14
0.7	0.67		0.15
0.8	0.72		0.13
0.9	0.79		0.09
1	0.85		0.04

connessione	media numero di passi	deviazione standard	numero di passi
0.4	1049		365
0.5	1351		506
0.6	1316		447
0.7	969		488
0.8	827		260
0.9	656		162
1	551		47

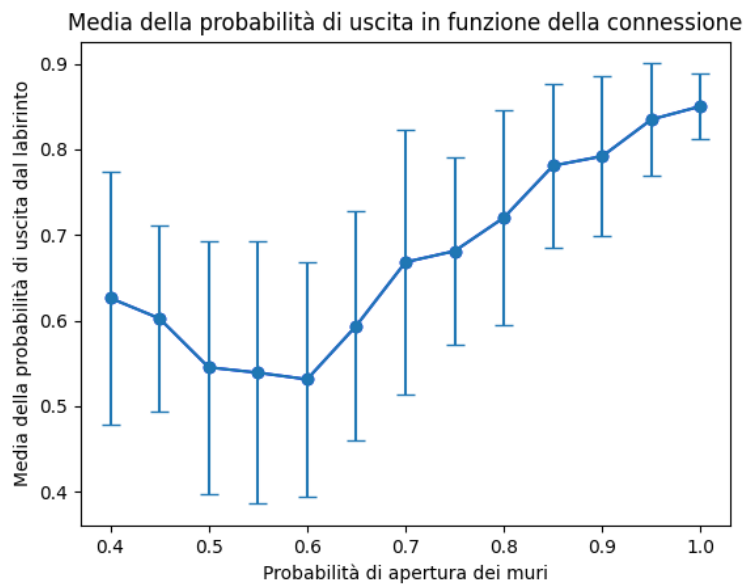


Figura 4.3: Media della probabilità di uscita da un labirinto entro 1000 passi in funzione della connessione, $d = 10$

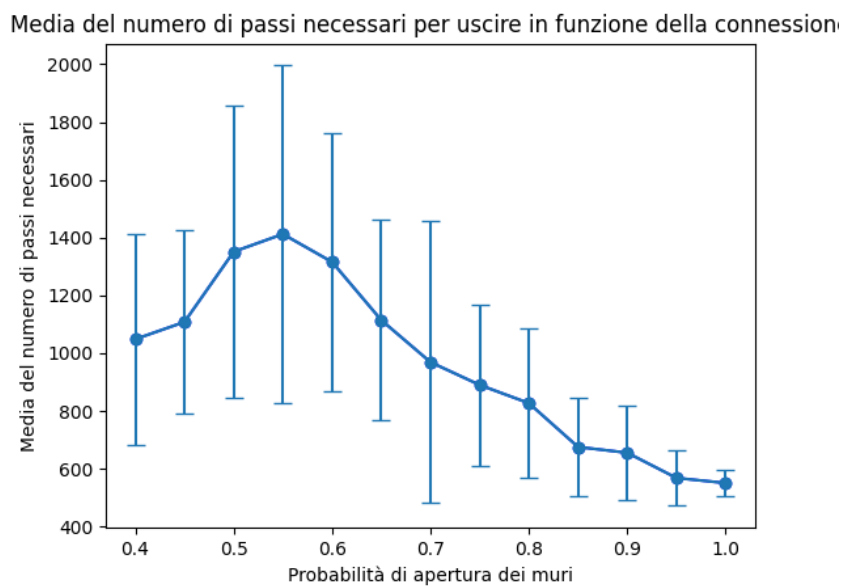


Figura 4.4: Media del numero di passi necessari ad uscire dal labirinto in funzione della connessione, $d = 10$

4.2.2 Discussione dei risultati

All'aumentare della connessione si osserva, nel complesso, un aumento della probabilità di uscita entro 1000 passi.

Quando la probabilità di apertura dei muri è bassa (0.4 – 0.6), il labirinto è più frammentato e il walker ha maggiori difficoltà a trovare un percorso verso l'uscita. Aumentando la connessione (da 0.7 in su), il labirinto diventa progressivamente più aperto e per il walker diventa più facile uscire.

Soprattutto per i valori bassi, le deviazioni standard sono molto alte; questo indica che i risultati delle simulazioni sono piuttosto variabili: in alcune configurazioni del labirinto il walker riesce a uscire facilmente, mentre in altre rimane intrappolato. Aumentando p , la deviazione standard cala ed il comportamento diventa più prevedibile.

Il numero di passi ha invece un picco attorno ai valori medi, dove il walker percorre in media più di 1300 passi. Con l'aumentare della connessione il numero medio di passi diminuisce; sono infatti possibili percorsi più diretti.

Le deviazioni standard sono elevate, specialmente per i valori intermedi. Per connessioni molto alte, si abbassano sensibilmente, segno che, in un labirinto molto aperto, i percorsi diventano molto regolari.

4.3 Confronto con random walk classico $2D$

4.3.1 Dati numerici e grafici

Le mappe prodotte rappresentano le distribuzioni stazionarie di due labirinti uguali dal punto di vista strutturale, ad esclusione della connessione:

```
open_walls_probabilities_heatmap = [0.5, 1]
```

I quadrati neri indicano le celle che il walker non ha mai visitato. Procedendo verso il bianco si trovano invece le celle visitate con frequenza crescente: tonalità più chiare corrispondono a un numero maggiore di visite.

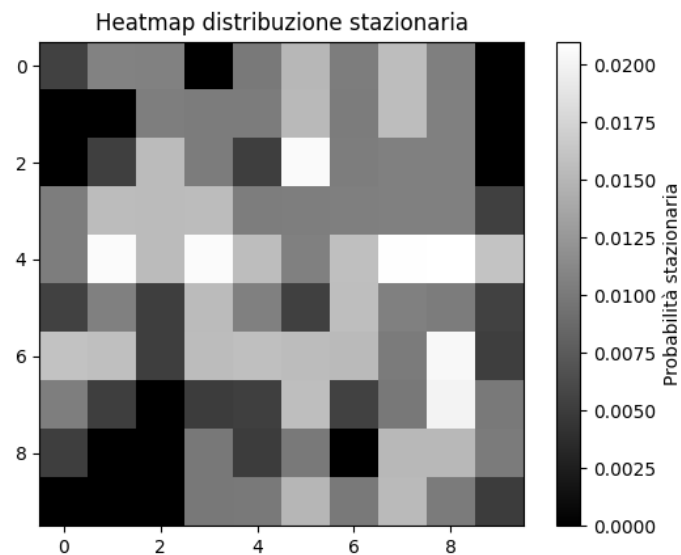


Figura 4.5: Distribuzione stazionaria di un labirinto 20×20 con $p = 0.5$

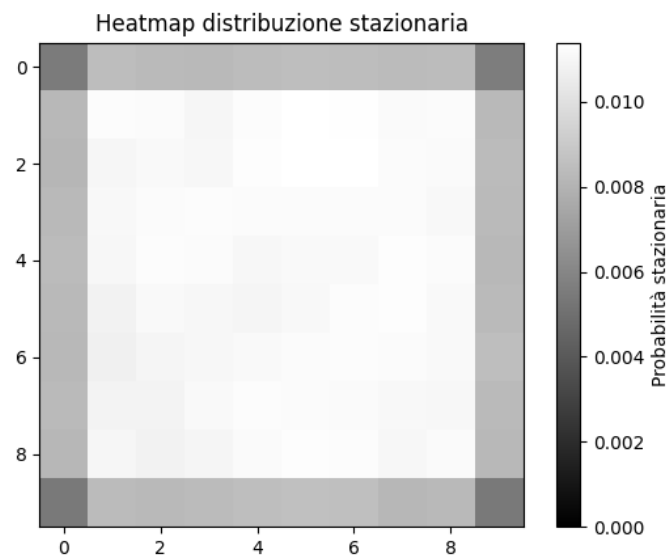


Figura 4.6: Distribuzione stazionaria di un labirinto 20×20 con $p = 1$

Distribuzione stazionaria di un labirinto 10×10 con $p = 0.5$:

0.028	0.028	0.0	0.0	0.0
0.084	0.056	0.0	0.0	0.0
0.083	0.055	0.0	0.028	0.028
0.055	0.111	0.083	0.083	0.056
0.028	0.056	0.055	0.056	0.028

Varianza distribuzione stazionaria per $p = 0.5$: 1.0555×10^{-3} .

Distribuzione stazionaria di un labirinto 10×10 con $p = 1$:

0.025	0.038	0.038	0.038	0.025
0.038	0.050	0.050	0.050	0.037
0.037	0.050	0.050	0.050	0.037
0.038	0.050	0.050	0.050	0.037
0.025	0.038	0.038	0.038	0.025

Varianza distribuzione stazionaria per $p = 1$: 7.4601×10^{-5} .

In entrambi i casi il walker ha percorso 1000000 di passi per ottenere la distribuzione stazionaria.

4.3.2 Discussione dei risultati

In entrambe le due configurazioni di labirinti (10×10 e 20×20), il confronto tra $p = 0.5$ e $p = 1$ evidenzia come la struttura del labirinto influenzi fortemente la distribuzione stazionaria:

- con connessione parziale la distribuzione è irregolare e alcune celle non vengono mai visitate;
- con connessione totale la distribuzione diventa molto più uniforme e tutte le celle risultano accessibili.

Quando $p = 1$, entrambe le distribuzioni stazionarie sono molto regolari, e le probabilità molto simili.

4.4 Specchi riflettenti

4.4.1 Dati numerici e grafici

Le tabelle seguenti contengono i dati numerici prodotti dalla simulazione python relativi alla probabilità di uscita dal labirinto 10×10 con $p = 0.7$ entro 1000 passi e al numero di passi necessari per uscire (media e deviazione standard) in funzione della probabilità di riflessione dei muri.

prob. di riflessione	media probabilità di uscita	deviazione standard probabilità di uscita
0.0	0.68	0.13
0.2	0.57	0.16
0.4	0.44	0.23
0.6	0.40	0.24
0.8	0.40	0.27
1	0.0	0.0

prob. di riflessione	media numero di passi	deviazione standard numero di passi
0.0	947	387
0.2	633	254
0.4	513	249
0.6	435	217
0.8	582	350
1	/	/

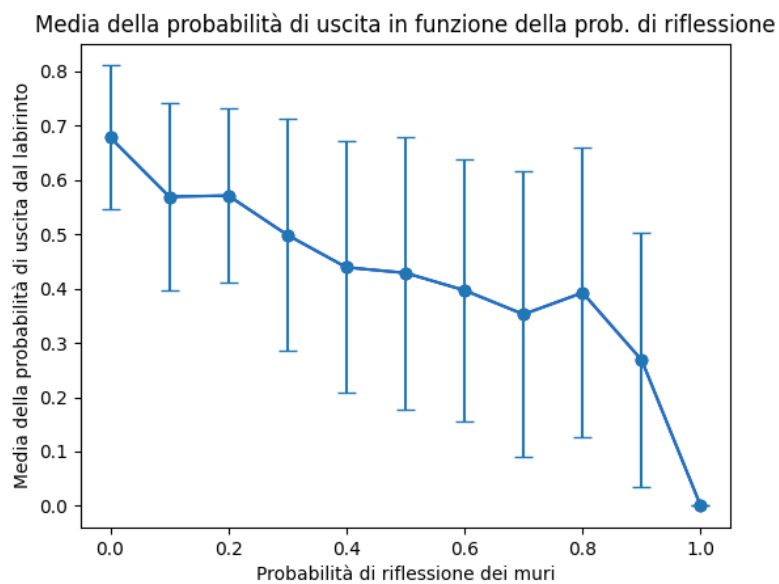


Figura 4.7: Media della probabilità di uscita da un labirinto entro 1000 passi in funzione della probabilità di riflessione, $d = 10$, $p = 0.7$

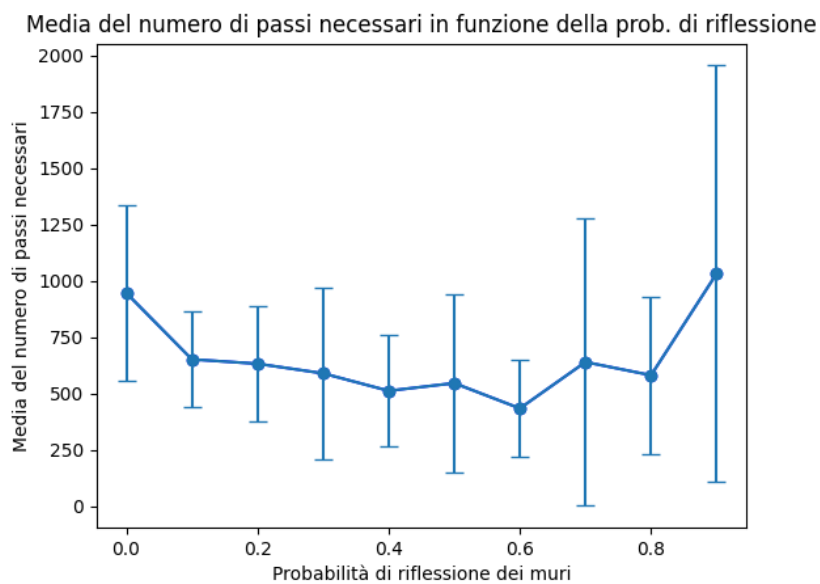


Figura 4.8: Media del numero di passi necessari ad uscire dal labirinto in funzione della probabilità di riflessione, $d = 10$, $p = 0.7$

4.4.2 Discussione dei risultati

All'aumentare della probabilità di riflessione si osserva una diminuzione della probabilità media di uscita. Anche la deviazione standard cresce con l'aumentare della riflessione: a causa delle riflessioni il percorso del walker dipende maggiormente dal caso, e alcune simulazioni portano all'uscita mentre altre no.

Nel caso limite $r = 1$, il walker viene sempre riflesso dai muri e non riesce mai a uscire, come indicato dalla probabilità media pari a 0 e dalla deviazione standard nulla.

Per valori piccoli o intermedi della riflessione il numero medio di passi diminuisce. Questo può essere spiegato dal fatto che, quando la riflessione aumenta, molte simulazioni non riescono a uscire entro 1000 passi e quindi la media dei passi viene calcolata solo sulle poche simulazioni che riescono ad uscire.

Le deviazioni standard sono relativamente elevate, segno che il tempo necessario per uscire può variare molto tra le simulazioni.

Dopo $r = 0.8$, il numero di passi e la deviazione standard aumentano notevolmente, indicando che il numero di passi diventa più disperso e meno prevedibile.

A $r = 1$, il numero di passi non è definito, poichè il walker non riesce mai ad uscire.

4.5 Violazione della proprietà di Markov

4.5.1 Dati numerici e grafici

Le tabelle seguenti contengono i dati numerici prodotti dalla simulazione python relativi alla probabilità di uscita dal labirinto entro 1000 passi e al numero di passi necessari per uscire (media e deviazione standard), sia dai labirinti ciechi che da quelli informati.

Labirinto 10×10 , $p = 0.7$:

tipologia	media probabilità di uscita	deviazione standard	probabilità di uscita
cieco	0.66		0.13
informato	0.96		0.04

tipologia	media numero di passi	deviazione standard	numero di passi
cieco	983		339
informato	310		90

Labirinto 20×20 , $p = 0.7$:

tipologia	media probabilità di uscita	deviazione standard	probabilità di uscita
cieco	0.10		0.05
informato	0.44		0.07

tipologia	media numero di passi	deviazione standard	numero di passi
cieco	3751		413
informato	1546		332

4.5.2 Discussione dei risultati

In entrambi i labirinti si osserva che l'informazione sulla direzione dell'uscita migliora significativamente la capacità di trovare il percorso corretto.

Prendendo ad esempio il labirinto 10×10 , si può notare che non solo il walker esce molto più spesso (da 0.66 a 0.96), ma ci mette anche molto meno (da 983 a 310 passi).

Il walker informato mostra anche una variabilità notevolmente minore fra i campioni, rispetto a quello cieco.

Quando la dimensione del labirinto aumenta, il problema diventa più difficile per entrambi i processi, ma la differenza tra i due tipi di walker rimane evidente.

4.6 Conclusioni

In questa tesi è stato studiato il comportamento di un random walker all'interno di labirinti generati casualmente mediante simulazioni numeriche in Python.

I risultati mostrano che la probabilità di uscita dal labirinto diminuisce all'aumentare della dimensione del labirinto, mentre il numero medio di passi necessari ad uscire cresce significativamente.

Anche la struttura del labirinto, in particolare il grado di connessione fra le celle, influisce in modo rilevante sul comportamento del walker.

Complicando ulteriormente la simulazione, tramite l'aggiunta di specchi riflettenti, si osserva un comportamento non monotono dei risultati al variare della probabilità di riflessione.

Il confronto tra walker cieco e walker informato evidenzia che l'utilizzo di informazione sulla direzione dell'uscita aumenta la probabilità di uscita e riduce il numero di passi necessari.

Nel complesso, questi risultati mostrano come modelli semplici di random walk possano essere utilizzati per studiare l'evoluzione nel tempo di complesse catene di Markov, evidenziando il ruolo della struttura del sistema e dei parametri del modello nel determinare la dinamica del processo.

Bibliografia

- [1] Carlo Sempi, *Introduzione alla probabilità*, Dipartimento di Matematica, Università del Salento, A.A. 2016-2017
- [2] S. Roch, *Introduction to Markov Chains*, Lecture Notes, University of Wisconsin–Madison (traduzione libera dell'autore)
- [3] Franco Flandoli, *catene1*, Dipartimento di matematica applicata UNIPI
- [4] G. R. Grimmett, M. V. Menshikov, S. E. Volkov, *Random Walks in Random Labyrinths*, Polymat, Moscow 1996
- [5] Michael Gold, *Python's Path Through Mazes: A Journey of Creation and Solution*, Medium, 2023
- [6] *Math Explorer's Club: The Mouse, the Maze and the Markov Chain*, Cornell University, 2008
- [7] Vincent Knight, Appendix: *Absorbing Markov Chains*, Game theory
- [8] Béla Bobbolas, *Random Graphs*, Cambridge University Press, 2001

Appendice A

Dimostrazione Teorema sezione 2.3

Teorema 33. *Una catena di Markov X_n irriducibile su uno spazio degli stati finito ha un'unica distribuzione stazionaria π .*

Se la catena non è solo irriducibile ma anche aperiodica, avremo per qualsiasi distribuzione iniziale ν ,

$$\lim_{n \rightarrow \infty} P^n \nu \{X_n = j\} = \pi(j) \quad \forall j \in S \quad (\text{A.1})$$

Dimostrazione. Supponiamo che siano soddisfatte le ipotesi del teorema, e sia P la matrice di probabilità di transizione della catena di Markov. Dimosteremo il teorema studiando l'azione di questa matrice sull'insieme \mathcal{P} delle distribuzioni di probabilità su S .

Ricordiamo dalla sezione 2.1.6 che, se ν^T è la distribuzione iniziale della catena di Markov, allora $\nu^T P^n$ è la distribuzione dopo n passi. Pertanto l'azione naturale della matrice di probabilità di transizione su \mathcal{P} è

$$\nu^T \rightarrow \nu^T P \quad (\text{A.2})$$

Si osservi che se ν^T è un vettore di probabilità, allora anche $\nu^T P$ lo è, infatti

$$(\nu^T P)_j = \sum_i \nu(i) p(i, j) \quad (\text{A.3})$$

e quindi

$$\sum_j (\nu^T P)_j = \sum_j \sum_i \nu(i) p(i, j) = \sum_i \nu(i) \sum_j p(i, j) = \sum_i \nu(i) \quad (\text{A.4})$$

l'ultima uguaglianza viene dalla proprietà di P secondo la quale tutte le somme sulle righe sono uguali a 1.

Questo implica che l'applicazione $\nu^T \rightarrow \nu^T P$ manda l'insieme \mathcal{P} in sé stesso.

Implica anche che (per induzione su n) P^n è una matrice stocastica per ogni $n = 1, 2, \dots$; viene dal fatto che ogni riga di P^n è un vettore di probabilità.

L' N -simpleso L'insieme \mathcal{P} è chiamato N -simpleso, dove N è la cardinalità dello spazio degli stati S : è il sottoinsieme di R^N ottenuto intersecando il primo ortante (l'insieme di tutti i valori con componenti non negative) con l'iperpiano costituito da tutti i vettori le cui componenti sommano a 1.

L' N -simpleso è un sottoinsieme chiuso e limitato di R^N . Di conseguenza, per il Teorema di Heine-Borel, esso è compatto.

L'argomento di Krylov–Bogoliubov Argomento che mostra che una distribuzione stazionaria esiste sempre. Mostra anche che esistono distribuzioni stazionarie anche per catene di Markov a stati finiti che non sono irriducibili.

L'argomento si basa sul fatto che il simpleso delle probabilità è compatto. Questo implica che esso possiede la proprietà di Bolzano–Weierstrass: ogni successione di vettori nel simpleso ha una sottosuccessione convergente.

Fissiamo un vettore di probabilità $\nu \in \mathcal{P}$, e consideriamo le cosiddette medie di Cesàro

$$\nu_n^T := \frac{1}{n} \sum_{k=1}^n \nu^T P^k \quad (\text{A.5})$$

Osserviamo che ciascuna media ν_n^T è un vettore di probabilità (perché una media di vettori di probabilità è sempre un vettore di probabilità), e quindi ogni ν_n^T appartiene al simpleso \mathcal{P} .

Di conseguenza, la successione ν_n^T ha una sottosuccessione convergente:

$$\lim_{k \rightarrow \infty} \nu_{n_k}^T = \pi^T \quad (\text{A.6})$$

Proposizione 34. *Il limite di qualunque sottosuccessione di ν_n^T è una distribuzione stazionaria per P .*

La dimostrazione di questo argomento si può trovare a pagina 6 di [2].

La metrica della variazione totale Per dimostrare l'unicità della distribuzione stazionaria sotto le ipotesi del Teorema 33, studieremo più attentamente l'azione della matrice di probabilità di transizione sul simpleso.

La metrica (funzione distanza) più naturale sul simpleso non è la solita distanza pitagorica (euclidea), ma piuttosto la metrica della variazione totale, detta anche distanza taxicab.

Essa è definita come segue: per due distribuzioni di probabilità $\nu, \mu \in \mathcal{P}$,

$$d(\mu, \nu) = \|\mu - \nu\|_{TV} := \frac{1}{2} \sum_{i \in S} |\nu(i) - \mu(i)| \quad (\text{A.7})$$

(il fattore $\frac{1}{2}$ è una convenzione, garantisce che la distanza non sia mai maggiore di 1).

Proposizione 35. *Supponiamo che tutte le entrate di P siano strettamente positive. Allora l'applicazione*

$$\nu^T \rightarrow \nu^T P \quad (\text{A.8})$$

è una contrazione stretta del semplice \mathcal{P} rispetto alla distanza di variazione totale. Cioè esiste $0 < \alpha < 1$ tale che per ogni coppia di vettori di probabilità μ, ν ,

$$\|\nu^T P - \mu^T P\|_{TV} \leq \alpha \|\nu^T - \mu^T\|_{TV} \quad (\text{A.9})$$

La dimostrazione di questa proposizione si può trovare a pagine 6-7 di [2].

Il teorema del punto fisso per applicazioni contrattive Cosa otteniamo dal sapere che l'azione della matrice di probabilità di transizione sul semplice è una contrazione?

Innanzitutto, ci dice che se iniziamo la catena di Markov con due distribuzioni iniziali diverse, allora le distribuzioni dopo un passo sono più vicine di quanto fossero all'inizio.

Di conseguenza, per induzione, dopo n passi sono ancora più vicine: infatti, la distanza in variazione totale diminuisce di un fattore α ad ogni passo, tendendo a zero esponenzialmente rapidamente quando $n \rightarrow \infty$.

Questo significa che la catena di Markov alla fine “dimentica” la distribuzione iniziale.

Segue un importante teorema sulle applicazioni contrattive che formalizza le conclusioni dell'argomento precedente.

Teorema 36. *Sia (S, d) uno spazio metrico compatto, e sia $F : S \rightarrow S$ una contrazione stretta, cioè una funzione tale che per un certo numero reale $\alpha < 1$,*

$$d(F(x), F(y)) \leq \alpha d(x, y) \quad \forall x, y \in S \quad (\text{A.10})$$

Allora F ha un unico punto fisso $z \in S$ (cioè un punto tale che $F(z) = z$), e l'orbita di ogni punto $x \in S$ converge a z , cioè, se F^n è l' n -esima iterata di F , allora

$$\lim_{n \rightarrow \infty} F^n(x) = z \quad (\text{A.11})$$

La dimostrazione di questo teorema si può trovare a pagina 8 di [2].

Dimostrazione del teorema 33 Per ora abbiamo mostrato che:

1. se la matrice di probabilità di transizione P ha entrate strettamente positive, allora l'applicazione

$$\nu^T \rightarrow \nu^T P \quad (\text{A.12})$$

è una contrazione stretta del semplice \mathcal{P} ;

2. una contrazione stretta di uno spazio metrico compatto ha un unico punto fisso;

3. tutte le orbite convergono al punto fisso.

Ne segue che, se P ha entrate strettamente positive, allora valgono tutte le conclusioni del Teorema 33.

Resta quindi da mostrare come eliminare l'ipotesi che le entrate di P siano strettamente positive.

Lemma 37. *Sia P la matrice di probabilità di transizione di una catena di Markov finita, irriducibile e aperiodica. Allora esiste un intero m tale che, per ogni $n \geq m$, la matrice P^n ha entrate strettamente positive.*

La dimostrazione di questo lemma si può trovare alle pagine 9-10 di [2].

Se P è aperiodica e irriducibile, come ipotizzato nel Teorema 33, allora esiste un intero $m \geq 1$ tale che

$$Q := P^m \tag{A.13}$$

ha entrate strettamente positive.

Le potenze di una matrice stocastica sono ancora matrici stocastiche, quindi Q soddisfa le ipotesi della Proposizione 35.

Pertanto, Q è strettamente contrattiva sul semplice \mathcal{P} .

Di conseguenza, per il Teorema del punto fisso per contrazioni, esiste un unico vettore di probabilità π^T tale che

$$\pi^T Q = \pi^T = \pi^T P^m, \tag{A.14}$$

e tale che per ogni $\nu^T \in \mathcal{P}$,

$$\lim_{n \rightarrow \infty} \nu^T P^n = \lim_{n \rightarrow \infty} \nu^T P^{nm} = \pi^T. \tag{A.15}$$

Quest'ultima affermazione vale per ν^T , come per $\nu^T P, \nu^T P^2, \dots$, poiché anch'essi sono vettori di probabilità.

Di conseguenza, per ogni $k = 0, 1, \dots, m - 1$,

$$\lim_{n \rightarrow \infty} \nu^T P^{nm+k} = \pi^T. \tag{A.16}$$

Ora, se una successione di vettori $\{v_n\}_{n \geq 1}$ ha la proprietà che le sottosuccessioni $\{v_{nm+k}\}_{n \geq 1}$, per $k = 0, 1, \dots, m - 1$, convergono tutte allo stesso limite w , allora l'intera successione converge a w .

Quindi,

$$\lim_{n \rightarrow \infty} \nu^T P^n = \pi^T. \tag{A.17}$$

Questo è equivalente all'enunciato del lemma 37.

Resta da mostrare che π è una distribuzione stazionaria, e che è l'unica distribuzione stazionaria.

Poniamo $\mu^T := \pi^T P$.

Allora $\mu^T P^m = \mu^T$.

Quindi μ è una distribuzione stazionaria per $Q = P^m$.

Ma π^T è l'unica distribuzione stazionaria per Q , poichè Q è strettamente contrattiva sul semplice.

Dunque $\mu = \pi$, e quindi π è una distribuzione stazionaria per P . Il fatto che sia l'unica distribuzione stazionaria segue analogamente.

□

Appendice B

Link al codice Python

<https://github.com/Maddy99799/labirinto-random-walk>