MARCELO DOS SANTOS

ALGORITMOS ALEATORIZADOS E MONTE CARLO VIA CADEIAS DE MARKOV PARA O CÁLCULO DA CENTRALIDADE DE INTERMEDIAÇÃO

Trabalho de Conclusão de Curso apresentado como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação na Universidade Federal do Paraná.

Orientador: Prof. Dr. André Luís Vignatti.
Coorientador: Prof. Dr. Murilo V. G. da Silva.

Curitiba
2021

AGRADECIMENTOS

Agradeço aos meus pais João e Inez e à minha irmã Laís pelo apoio e incentivo.

Ao professor André Vignatti pela orientação deste trabalho.

Ao professor Murilo da Silva pela coorientação.

Aos meus colegas do curso de ciência da computação.

A todos que, de uma forma ou de outra, contribuíram para a realização deste trabalho.

RESUMO

Este trabalho apresenta um estudo geral a respeito da teoria de cadeias de Markov, método de Monte Carlo e algumas aplicações de algoritmos aleatorizados. Inicialmente é apresentada a teoria de cadeias de Markov juntamente com algumas definições e conceitos que são necessários para modelar a aleatoriedade. Também será mostrado o método de Monte Carlo juntamente com o método Monte Carlo via cadeias de Markov, onde será dado ênfase ao algoritmo *Metropolis*. Será mostrada a teoria utilizada para limitar quantitativamente o tempo de mistura em cadeias de Markov, pois em qualquer aplicação deve-se ter garantias de que o algoritmo vai executar em tempo polinomial em função da entrada. Ao longo do trabalho serão mostradas algumas aplicações de algoritmos aleatorizados e no último capítulo o algoritmo *Metropolis* é utilizado para o cálculo da centralidade de intermediação, que é uma medida frequentemente utilizada na área de redes complexas.

SUMÁRIO

CAPÍTULO 1

INTRODUÇÃO

Neste trabalho será mostrada a teoria de cadeias de Markov e método de Monte Carlo, juntamente com algumas aplicações simples e uma aplicação final para o cálculo da centralidade de intermediação, que é uma medida centralidade utilizada em redes complexas.

Infelizmente muitos problemas práticos possuem até o momento somente algoritmos com tempo de execução exponencial[1]. Assim muitas vezes se torna inviável resolver estes problemas de forma exata. Uma classe de problemas que aparentemente são difíceis de serem calculados com exatidão são os de *contagem*, pertencentes à classe #P [1]. Como muitos destes problemas possuem diversas aplicações, é necessário encontrar alternativas para resolver estes problemas, mesmo que de forma aproximada, mas que executem em tempo polinomial. Uma das alternativas existentes é encontrar algoritmos aleatorizados que executam em tempo polinomial. Apesar da solução não ser exata, em muitos casos é possível saber de antemão qual é o erro fornecido pelo algoritmo.

No nosso contexto surge a necessidade de primeiro entender os métodos que modelam processos aleatórios. Para isso, neste trabalho será feito inicialmente um estudo de cadeias de Markov de tempo discreto.

Para modelar o comportamento de um sistema, determinamos os estados possíveis que este sistema pode ocupar e indicamos como o sistema transiciona entre estes estados. Considera-se que a cada momento o sistema está em apenas um estado. Assim, um processo estocástico são transições que ocorrem em uma família de estados (variáveis aleatórias) e estas transições indicam a evolução do sistema com o tempo. Uma cadeia de Markov é um caso particular de um processo estocástico. Matematicamente, a cadeia de Markov é composta por um conjunto de estados Ω por onde a cadeia transiciona ao longo do tempo, e uma matriz estocástica que fornece a probabilidade de transição entre os estados. Na cadeia de Markov, o próximo estado da cadeia depende apenas do estado atual. É como se a cadeia "esquecesse" toda sua história no tempo, pois ela considera apenas seu estado anterior para efetuar transições [2].

Para cada tempo t é possível determinar a probabilidade de um estado $x \in \Omega$ ser alcançado e sob determinadas condições a distribuição de probabilidades da cadeia convergirá assintoticamente para uma distribuição estacionária π quando $t \to \infty$.

[1]Se o tamanho da entrada do algoritmo é n, então tempo exponencial indica que o tempo de execução do algoritmo cresce exponencialmente com n. Da mesmo forma, tempo polinomial indica que o tempo de execução do algoritmo cresce polinomialmente com n.

Além de entender cadeias de Markov, também é necessário entender como utilizar a aleatoriedade para resolver problemas e para isso utilizaremos o método de Monte Carlo. O método de Monte Carlo é uma ampla classe de algoritmos computacionais de amostragem e simulação que possibilita estimar a solução de uma variedade de problemas.

Existem muitas variações de método de Monte Carlo, mas eles normalmente seguem o padrão:

- Defina um domínio de entrada.
- Dado o domínio, gere dados aleatoriamente de uma distribuição de probabilidade.
- Realize um cálculo determinístico sobre os dados gerados.
- Agregue os resultados e gere a saída

Neste método mais simples de Monte Carlo, estamos realizando o que é conhecido como amostragem direta, onde cada amostra aleatória é gerada de forma independente, seguindo uma distribuição de probabilidades. Outra forma de gerar amostras aleatoriamente seguindo uma distribuição de probabilidades é por meio de uma cadeia de Markov, onde as amostras geradas possuem certa correlação, neste caso temos o que é conhecido como amostragem por cadeia de Markov, que origina o *método de Monte Carlo via Cadeias de Markov* [3].

O método de Monte Carlo via Cadeias de Markov (MCMC) é uma grande classe de algoritmos de amostragem e fornece um procedimento geral para a solução de diversos problemas computacionais. Dentre os principais algoritmos de amostragem desta classe pode-se citar a amostragem de Gibbs, *importance sampling* e *Metropolis - Hasting*. Estes algoritmos desempenharam um papel significante em áreas como estatística, econometria, física e ciência da computação e ainda continuam tendo diversas aplicações. O algoritmo *Metropolis* está entre os 10 principais algoritmos com maior influência no desenvolvimento da ciência e da engenharia no século 20 [4].

O método MCMC é uma ferramenta para a seguinte tarefa computacional: seja Ω um conjunto de estados muito grande, mas finito, de estruturas combinatórias como por exemplo o conjunto de todas as combinações possíveis de um sistema físico ou o conjunto de todas as soluções para um problema de otimização. Seja π uma distribuição de probabilidades em Ω. O objetivo é amostrar elementos $x \in \Omega$ com probabilidade $\pi(x)$ [5].

Dentre as principais aplicações do método MCMC, pode-se citar:

- Problemas de contagem: Estimar a cardinalidade de Ω. Por exemplo, dado um grafo G, Ω pode ser um conjunto que contém todos os conjuntos independentes de G e π seria a probabilidade de amostrar cada um destes conjuntos independentes. A estimativa de $|\Omega|$ pode ser obtida amostrando conjuntos independentes $\Omega(G_i)$, onde G_i é um subgrafo de G com i arestas. Por meio de um procedimento simples calcula-se a razão $|\Omega(G_i)|/|\Omega(G_{i-1})|$ onde i varia de 1 até o número de arestas. O valor final de $|\Omega|$ é obtido pelo produto telescópico de todas estas razões (ver exemplo 3.3).
- Física estatística: Neste caso Ω é um conjunto de todas as configurações possíveis de um sistema físico e π é a uma distribuição de probabilidades que relaciona a energia de um estado com a probabilidade de sua ocorrência. O objetivo aqui é amostrar configurações significativas para calcular alguma propriedade do sistema como energia média.
- Otimização combinatória: Neste caso Ω é o conjunto de soluções de um problema de otimização e π é uma distribuição de probabilidades construída de maneira que os estados que possuam um valor

melhor, de acordo com uma função objetivo, tenham uma probabilidade maior de ocorrência no processo de amostragem. Esta abordagem é a base do algoritmo heurístico popularmente conhecido como *simulated annealing*.

O método MCMC possui também aplicações em estatística Bayesiana [6]:

- Dado a função a priori e a função de verossimilhança, usa-se MCMC para calcular a posteriori do modelo.
- Estimar o valor esperado de uma função $f(x)$ de acordo com a probabilidade $p(x)$:

$$\mathbf{E}_{p(x)}[f(x)] = \int f(x)p(x)dx. \tag{1.1}$$

Vale ainda ressaltar que para certos problemas multidimensionais como calcular o volume de um corpo convexo em d dimensões, MCMC é a única abordagem que fornece uma solução em tempo razoável (polinomial em d) [6].

Para realizar a amostragem, o método MCMC realiza uma caminhada aleatória em um espaço de estados Ω. As transições entre os estados da cadeia ocorrem seguindo uma pequena perturbação nos elementos $x \in \Omega$, de forma que seja fácil e eficiente de simular. Para realizar a amostragem, iniciamos o processo em um estado arbitrário de Ω e simulamos a cadeia de Markov um número τ de passos. A cadeia deve ser ergódica, isso quer dizer intuitivamente que para τ grande o suficiente, o histograma dos estados amostrados na caminhada aleatória é sempre proporcional à distribuição π desejada e independente do estado inicial.

Em muitos casos, não é difícil construir uma cadeia de Markov, mas o que não sabemos exatamente é o número de passos τ do algoritmo. Para que o algoritmo seja eficiente, devemos ter um τ muito menor que $|\Omega|$. Por exemplo, se estamos fazendo a simulação de um sistema físico com n partículas ou se estamos resolvendo um problema computacional onde a entrada são n bits, e se $|\Omega|$ é proporcional a 2^n ($\exists\, c > 0$ tal que $|\Omega| = c \cdot 2^n$), então τ deve ser uma função com crescimento mais lento que 2^n, como por exemplo polinomial em n. Se não for, não faz sentido usar uma cadeia de Markov para realizar este tipo de simulação. O tempo que a cadeia demora para ficar próxima da distribuição estacionária (diremos que as duas distribuições estarão no máximo $\epsilon-$distantes uma da outra) é chamado tempo de mistura. Quando a cadeia realiza uma curta caminhada aleatória em Ω (τ é uma função com crescimento lento em n quando comparado com $|\Omega|$) e sua distribuição de probabilidades está muito próxima de probabilidade estacionária, então dizemos que a cadeia possui um rápido tempo de mistura [5, 7].

Em muitos casos é difícil provar que a cadeia de Markov possui um rápido tempo de mistura. Existem diversas técnicas que são usadas para criar limitantes para o tempo de mistura da cadeia, dentre elas pode-se citar: método do acoplamento, acoplamento de caminhos, e outras técnicas baseadas em métodos espectrais como condutância da cadeia e congestionamento [7]. Assim, neste trabalho também serão mostradas algumas técnicas comumente utilizadas para limitar o tempo de mistura de cadeias de Markov.

Este trabalho está organizado da seguinte maneira: no Capítulo 2 é fornecida a teoria de cadeias de Markov juntamente com os tipos de estados e suas classificações, cálculo da distribuição estacionária, definição matemática de tempo de mistura e outras propriedades que servirão de base para estabelecer o método MCMC.

No Capítulo 3, será mostrado o método de Monte Carlo com alguns exemplos simples de aplicação, seguido do método MCMC com ênfase ao algoritmo *Metropolis-Hasting*, e por último serão mostradas definições de algoritmos de aproximação e amostragem que executam em tempo polinomial (FPRAS e FPAUS).

No Capítulo 4, serão mostradas duas técnicas muito conhecidas para limitar o tempo de mistura de uma cadeia de Markov, a do acoplamento e da condutância.

Por último, no Capítulo 5, o algoritmo *Metropolis* é aplicado no cálculo da centralidade de intermediação.

Para que o texto não fique muito sobrecarregado, muitas das demonstrações serão omitidas ao longo do texto e mostradas com mais detalhes nos apêndices ou é indicado uma referência com os detalhes.

CAPÍTULO 2

CADEIAS DE MARKOV

Um processo estocástico é definido como uma coleção de variáveis aleatórias $\mathbf{X} = \{X(t)|t \in T\}$. Estas variáveis são indexadas por t, que geralmente indica o tempo. A variável $X(t)$ (que denotaremos por X_t) é o estado do processo no tempo t e normalmente está associado com a evolução temporal de algum sistema como por exemplo flutuações no mercado de ações, contagem de bactérias, movimentos de uma partícula em um gás. Se T é um conjunto infinito e contável, dizemos que $\mathbf{X}$ é um processo de tempo discreto.

Definição 2.1. *Um processo estocástico de tempo discreto* $(X_0, X_1, \dots)$ *é uma cadeia de Markov se*

$$\mathsf{Pr}(X_t = a_t|X_{t-1} = a_{t-1}, X_{t-2} = a_{t-2}, \dots, X_0 = a_0) = \mathsf{Pr}(X_t = a_t|X_{t-1} = a_{t-1}), \tag{2.1}$$

onde a_i *é um estado qualquer.*

Esta definição significa que o estado do sistema no tempo t depende apenas do estado do sistema no tempo $t-1$ e é independente da trajetória do sistema do tempo inicial até o tempo $t-1$. Esta propriedade é conhecida como *propriedade de Markov*.

Vamos considerar que o espaço de estados $\Omega = \{1, 2, \dots\}$ e o tempo t é discreto e indexado por números inteiros. Como o sistema evolui no tempo, estamos interessados na probabilidade do sistema transicionar do estado i para o estado j em um passo (tempo $t-1$ para o tempo t). Esta probabilidade é dada por

$$P_{i,j} = \mathsf{Pr}(X_t = j|X_{t-1} = i) \tag{2.2}$$

Com isso, pode-se definir uma cadeia de Markov com a matriz de transição (também conhecida como matriz estocástica) de um único passo

$$\mathbf{P} = \begin{pmatrix} P_{1,1} & P_{1,2} & \cdots & P_{1,j} & \cdots \\ P_{2,1} & P_{2,1} & \cdots & P_{2,j} & \cdots \\ \vdots & \vdots & \ddots & \vdots & \\ P_{i,1} & P_{i,2} & \cdots & P_{i,j} & \cdots \\ \vdots & \vdots & & \vdots & \ddots \end{pmatrix}$$

onde cada elemento $P_{i,j}$ denota a probabilidade de transição do estado i para j. Devemos ter que $\sum_j P_{i,j} = 1$, pois a probabilidade de transição de i para qualquer estado é 1.

Seja $p_i(t)$ a probabilidade do sistema estar no estado i no tempo t. Pode-se definir um vetor de distribuição de probabilidades $\mathbf{p}(t) = (p_1(t), \ldots, p_n(t))$. Para que $\mathbf{p}(t)$ seja uma distribuição de probabilidade, devemos ter

$$\sum_i p_i(t) = 1. \tag{2.3}$$

Para se calcular a probabilidade do sistema estar no estado i no tempo t, deve-se considerar a probabilidade do sistema estar no estado j no tempo $t-1$ e multiplicar pela probabilidade de transição de j para i. Como existem vários j possíveis, deve-se efetuar a soma sobre todos eles, ou seja:

$$p_i(t) = \sum_j p_j(t-1)P_{j,i}, \tag{2.4}$$

escrevendo na forma de produto de matrizes temos

$$\mathbf{p}(t) = \mathbf{p}(t-1)\mathbf{P}. \tag{2.5}$$

Por indução em m ($m \geq 0$), pode-se provar que

$$\mathbf{p}(t+m) = \mathbf{p}(t)\mathbf{P}^m \tag{2.6}$$

e o elemento

$$P_{i,j}^m = \mathsf{Pr}(X_{t+m} = j | X_t = i). \tag{2.7}$$

indica a probabilidade do sistema transicionar de i para j em exatamente m passos.

Exemplo 2.1. *Cadeia de Markov com 2 estados*

Considere um sapo que mora numa lagoa com duas pedras A e B. O sapo sempre está em cima de uma das duas pedras. A cada momento, o sapo pode continuar na mesma pedra ou pular para a outra pedra.

Considere que a matriz de transição (troca de pedra) é dada por

$$\mathbf{P} = \begin{pmatrix} P_{A,A} & P_{A,B} \\ P_{B,A} & P_{B,B} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1/2 & 1/2 \end{pmatrix}$$

Desejamos saber qual é a pedra mais provável de encontrar o sapo após um longo tempo de evolução dessa cadeia.

Considere que inicialmente o sistema encontra-se no estado A, ou seja, $\mathbf{p}(0) = (1 \;\; 0)$. Assim pela equação (2.6), o estado do sistema nos tempos posteriores é $\mathbf{p}(t) = \mathbf{p}(0)\mathbf{P}^t$. Para $t = 1, 2, 3, 4$, obtemos as distribuições $\mathbf{p}(t)$ como sendo, respectivamente

$$\begin{pmatrix} 0 & 1 \end{pmatrix}, \begin{pmatrix} 1/2 & 1/2 \end{pmatrix}, \begin{pmatrix} 1/4 & 3/4 \end{pmatrix} \text{ e } \begin{pmatrix} 3/8 & 5/8 \end{pmatrix}.$$

Para $t = 10$, temos

$$\mathbf{p}(t = 10) = \begin{pmatrix} 0.33398 & 0.66602 \end{pmatrix}.$$

Por outro lado, se o estado inicial do sistema fosse B, ou seja, $\mathbf{p}(0) = (0 \;\; 1)$, a distribuição de probabilidades

$\mathbf{p}'(t)$ para $t = 10$ seria:

$$\mathbf{p}'(t = 10) = \begin{pmatrix} 0.33301 & 0.66699 \end{pmatrix}.$$

De fato essas duas distribuições são muito próximas. Uma forma de quantificar isso é pela distância euclidiana entre elas:

$$\|\mathbf{p}(t = 10) - \mathbf{p}'(t = 10)\| = \|(1\ \ 0)P^{10} - (0\ \ 1)P^{10}\| = 0.0013811. \tag{2.8}$$

Como será visto mais adiante, para alguns tipos específicos de cadeias de Markov, o estado do sistema converge para uma distribuição estacionária quando $t \to \infty$. Neste caso, a distância calculada acima tende a zero. No caso da cadeia deste exemplo, a distribuição estacionária é $\mathbf{p}(\infty) = (1/3\ \ 2/3)$.

2.1 Classificação dos estados e cadeias

Definição 2.2. *Sejam $i, j \in \Omega$. O estado j é acessível a partir do estado i se existe algum inteiro $t \geq 0$ tal que $P^t_{i,j} > 0$. Se i e j são ambos acessíveis um a partir do outro, dizemos que i e j são comunicantes ou se comunicam e denotamos $i \leftrightarrow j$.*

A relação de comunicação define uma relação de equivalência, pois a relação $\leftrightarrow$ é:

1. *Reflexiva*: $\forall i \in \Omega$, $i \leftrightarrow i$.
2. *Simétrica*: $i \leftrightarrow j \Rightarrow j \leftrightarrow i$
3. *Transitiva*: $i \leftrightarrow j$ e $j \leftrightarrow k \Rightarrow i \leftrightarrow k$.

Lembrando da definição de classes de equivalência:

Definição 2.3. *Seja R uma relação de equivalência em um conjunto $A \neq \varnothing$. A classe de equivalência de um elemento $a \in A$, com respeito a relação R é o conjunto de todos os elementos de A que estão relacionados com a e será denotado por $[a]_R$. Assim*

$$[a]_R = \{x \in A | xRa\}. \tag{2.9}$$

Portanto a relação de comunicação particiona o espaço de estados Ω em classes de equivalência disjuntas ou classes comunicantes.

Definição 2.4. *Uma cadeia é irredutível se $\forall i, j \in \Omega$, $\exists t \geq 0$ tal que $P^t_{i,j} > 0$. Isto significa que a partir de qualquer estado é possível chegar a qualquer outro estado da cadeia. Neste caso existe apenas uma classe de equivalência ou classe comunicante.*

Uma cadeia de Markov pode ser representada por um grafo direcionado, onde os vértices são estados e os arcos são as probabilidades de transição. Se esta representação for utilizada, então dizer que um estado j é alcançável a partir de i é equivalente a dizer que existe um caminho direcionado de i a j. Além disso, dizer que uma cadeia é irredutível é equivalente a dizer que o grafo que representa a cadeia é fortemente conexo.

Exemplo 2.2. *Cadeia de Markov redutível*

Considere uma cadeia de Markov com a seguinte matriz de transição:

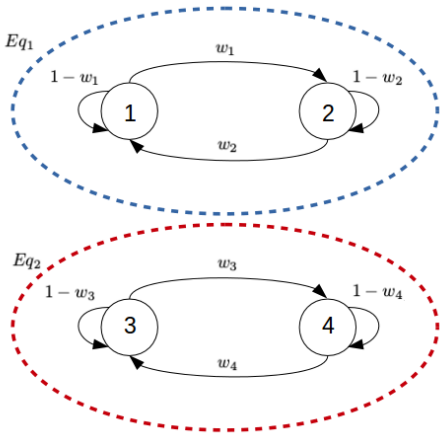

Figura 2.1: Grafo que representa a cadeia $\mathbf{P}_r$. As linhas pontilhadas delimitam as duas classes de equivalência Eq_1 e Eq_2. Se a cadeia está em um estado $i \in Eq_1$, a probabilidade de ocorrer a transição para um estado $j \in Eq_2$ em um tempo t é $P^t_{i,j} = 0$, $\forall t \geq 0$.

$$\mathbf{P}_r = \begin{pmatrix} 1-w_1 & w_1 & 0 & 0 \\ w_2 & 1-w_2 & 0 & 0 \\ 0 & 0 & 1-w_3 & w_3 \\ 0 & 0 & w_4 & 1-w_4 \end{pmatrix},$$

onde w_1, w_2, w_3 e w_4 são números reais entre 0 e 1. Esta cadeia pode ser representada pelo grafo mostrado na figura (2.1).

Analisando a matriz de transição $\mathbf{P}_r$ percebe-se que se o estado inicial da cadeia é o estado 1 ou 2, a cadeia permanece transicionando entre estes dois estados para tempos posteriores, pois a probabilidade de ocorrer transição dos estados 1 ou 2 para 3 ou 4 é nula. O mesmo processo ocorre se os estados iniciais forem 3 ou 4. Nesta caso nunca ocorrerá um transição para os estados 1 ou 2. Portanto esta cadeia é redutível.

Vamos agora fazer a distinção entre estados transientes e estados recorrentes. Seja $r^t_{i,j}$ a probabilidade da primeira transição de i para j ocorrer no tempo t, ou seja,

$$r^t_{i,j} = \mathsf{Pr}(X_t = j \,\text{e para}\, 1 \leq s \leq t-1, X_s \neq j | X_0 = i). \tag{2.10}$$

Na equação acima, se realizarmos uma soma em t de 0 a ∞ e fizermos $i = j$, temos a probabilidade de sair do estado i e ocorrer uma transição de retorno para i em um tempo posterior qualquer. Assim podemos definir estado recorrente:

Definição 2.5. *Um estado i é recorrente se $\sum_{t\geq 1} r^t_{i,i} = 1$. Se um estado i não é recorrente, então ele é transiente. Uma cadeia de Markov é recorrente se todo estado da cadeia é recorrente.*

Vamos denotar por $h_{i,j}$ o tempo esperado para a cadeia alcançar o estado j pela primeira vez partindo de i, ou seja

$$h_{i,j} = \sum_{t>0} t\, r^t_{i,j}. \tag{2.11}$$

Assim $h_{i,i}$ denota o tempo para a cadeia voltar para o estado i partindo de i. Este tempo também é chamado de tempo de recorrência.

Definição 2.6. *Um estado recorrente i é recorrente positivo se $h_{i,i} < \infty$. Caso contrário, ele é de recorrência nula.*

Definição 2.7. *Um estado j em uma cadeia de Markov de tempo discreto é periódico se existe um inteiro $\Delta > 1$ tal que $\mathsf{Pr}(X_{t+s} = j|X_t = j) = 0$, a menos que s seja divisível por Δ. Uma cadeia de Markov de tempo discreto é periódica se todo estado $j \in \Omega$ da cadeia for periódico. Uma cadeia que não é periódica é aperiódica.*

Exemplo 2.3. *Caminhada aleatória no $\mathbb{Z}_{10}$*

Para exemplificar o conceito de aperiodicidade, considere o exemplo de uma caminhada aleatória no $\mathbb{Z}_{10}$. Se no tempo t a cadeia encontra-se no estado $i \mod 10$, no tempo $t+1$ o estado da cadeia será $(i+1) \mod 10$ com probabilidade $1/2$ ou $(i-1) \mod 10$, também com probabilidade $1/2$. É fácil de ver que caso a cadeia comece no estado 0, após um número par de passos a cadeia estará num estado par e após um número ímpar de passos a cadeia estará num estado ímpar, ver figura (2.2). Este é um exemplo de comportamento periódico. Perceba que $\forall j \in \mathbb{Z}_{10}$, $\mathsf{Pr}(X_{t+s} = j|X_t = j) = 0$ a menos que s seja divisível por 2.

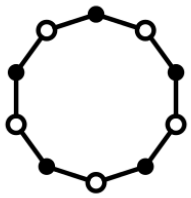

Figura 2.2: Caminhada aleatória no $\mathbb{Z}_{10}$ é periódica. Vértices pretos e brancos indicam que existem estados pares e ímpares.

Definição 2.8. *Um estado que é recorrente positivo e aperiódico é um estado ergódico. Uma cadeia de Markov é ergódica se todos seus estados são ergódicos.*

2.2 Distribuição estacionária

Como foi visto anteriormente, a distribuição de probabilidade no tempo $t+1$ pode ser obtida por meio da equação

$$\mathbf{p}(t+1) = \mathbf{p}(t)\mathbf{P}. \tag{2.12}$$

Estamos agora interessados no caso em que a cadeia de Markov atinge um estado estacionário, ou seja, a distribuição de probabilidades não é mais alterada após a aplicação da matriz de transição. Neste caso $\mathbf{p}(t+1) = \mathbf{p}(t)$ é chamada de distribuição estacionária ou distribuição de equilíbrio e será denotada por $\bar{\pi}$.

Definição 2.9. *Uma distribuição estacionária de uma cadeia de Markov é uma distribuição de probabilidade $\bar{\pi}$ tal que*

$$\bar{\pi} = \bar{\pi}\,\mathbf{P}. \tag{2.13}$$

Teorema 2.1. *Qualquer cadeia de Markov ergódica, finita e irredutível possui as propriedades:*

1. *A cadeia possui uma única distribuição estacionária $\bar{\pi} = (\pi_0, \ldots, \pi_n)$;*

2. $\forall i, j \in \Omega$*, o limite* $\lim_{t\to\infty} P^t_{j,i}$ *existe e é independente de* j*;*

3. $\pi_i = \lim_{t\to\infty} P^t_{j,i} = 1/h_{i,i}$*.*

Prova: Ver [1]. □

No exemplo 2.1 (exemplo do sapo) é facil verificar que $\bar{\pi} = (1/3, 2/3)$ satisfaz a equação (2.13) acima. Uma pergunta que surge é quanto tempo a cadeia de Markov demora para alcançar o estado estacionário. Para responder isto, vamos primeiro desenvolver a intuição com o exemplo seguinte e depois ver a Proposição 2.1 que responde esta pergunta para o caso mais geral de uma cadeia de Markov com um número arbitrário de estados.

Exemplo 2.4. *Cadeia de Markov com dois estados*

Considere o caso mais geral de uma cadeia de Markov com 2 estados. No exemplo 2.1 teríamos uma probabilidade de $1-p$ do sapo permanecer na pedra A e p de transicionar da pedra A para a pedra B. A probabilidade de permanecer na pedra B é $1-q$ e q de transicionar da pedra B para A. Vamos encontrar a distribuição de probabilidades para um tempo t qualquer. A matriz de transição que modela qualquer cadeia com 2 estados é

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

Observe que os elementos de qualquer linha somam 1.

Dada a matriz de transição acima, nosso objetivo é encontrar uma distribuição $\bar{\pi}$ que satisfaça a equação (2.13). Aplicando a transposta na equação (2.13) obtemos

$$\mathbf{P}^\top \mathbf{x} = \lambda \mathbf{x}, \tag{2.14}$$

onde $\mathbf{x} = \bar{\pi}^\top$ e $\lambda = 1$. A equação (2.14) possui a forma de uma equação de autovalor (ver apêndice A) do tipo

$$\mathbf{A}\mathbf{v}_i = \lambda_i \mathbf{v}_i, \tag{2.15}$$

onde $\mathbf{v}_i$ é $i-$ésimo autovetor de $\mathbf{A}$ e λ_i seu respectivo autovalor[1].

Para analisar a convergência para o estado estacionário desta cadeia de Markov, será necessário encontrar os autovalores da matriz $\mathbf{P}^\top$. O teorema de *Perron-Frobenius* [8, 9] garante que $\mathbf{P}^\top$ possui apenas um autovalor $\lambda_1 = 1$, cujo respectivo autovetor $\mathbf{v}_1 = \bar{\pi}^\top$. O outro autovalor possui valor absoluto menor que 1.

Considerando $\mathbf{A} = \mathbf{P}^\top$ e $\mathbf{x} = \mathbf{v}_i$ na equação (2.15), podemos escrever:

$$\mathbf{P}^\top \mathbf{v}_i = \lambda \mathbf{v}_i \Rightarrow (\mathbf{P}^\top - \lambda \mathbb{1})\mathbf{v}_i = 0. \tag{2.16}$$

A equação acima é equivalente a (ver Teorema 8.2.2. de [10])

$$\det(\mathbf{P}^\top - \lambda \mathbb{1}) = 0 \tag{2.17}$$

ou

$$\begin{vmatrix} 1-p-\lambda & q \\ p & 1-q-\lambda \end{vmatrix} = 0$$

[1] A equação (2.13) é uma equação de auto-valor à esquerda.

que fornece $\lambda_1 = 1$ (como esperado pelo teorema de *Perron-Frobenius*) e $\lambda_2 = 1 - q - p$. Vamos agora encontrar $\mathbf{v}_1$ e $\mathbf{v}_2$:

$$\mathbf{P}^\top \mathbf{v}_1 = 1\mathbf{v}_1 \quad \text{e} \quad \mathbf{P}^\top \mathbf{v}_2 = (1-q-p)\,\mathbf{v}_2 \tag{2.18}$$

considerando

$$\mathbf{v}_1 = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \quad \text{e} \quad \mathbf{v}_2 = \begin{pmatrix} w_1 \\ w_2 \end{pmatrix},$$

temos os sistemas de equações

$$\begin{pmatrix} 1-p & q \\ p & 1-q \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \quad \text{e} \quad \begin{pmatrix} 1-p & q \\ p & 1-q \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = (1-q-p) \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}.$$

A solução encontrada é:

$$\mathbf{v}_1 = \bar{\pi}^\top = \frac{1}{p+q} \begin{pmatrix} q \\ p \end{pmatrix} \quad \text{e} \quad \mathbf{v}_2 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}. \tag{2.19}$$

Vamos agora finalmente analisar o quão rápido a cadeia converge para o estado $\bar{\pi}$.

Considere que inicialmente a cadeia encontra-se no estado $\mathbf{p}(0) = (\alpha, 1-\alpha)$, onde α é um parâmetro. Se $\alpha = 0$, então $\mathbf{p}(0) = (0,1)$ e se e $\alpha = 1$, então $\mathbf{p}(0) = (1,0)$. Assim estamos modelando qualquer estado inicial possível. Observe que $\mathbf{p}(0)$ pode ser reescrito em termos dos autovetores da matriz $\mathbf{P}^\top$ como

$$\mathbf{p}(0) = \mathbf{v}_1^\top + \left(\alpha - \frac{q}{p+q}\right)\mathbf{v}_2^\top = \bar{\pi} + \left(\alpha - \frac{q}{p+q}\right)\mathbf{v}_2^\top. \tag{2.20}$$

O estado da cadeia em um tempo t posterior é

$$\begin{aligned}
\mathbf{p}(t) &= \mathbf{p}(0)\mathbf{P}^t && (2.21)\\
&= \left[\mathbf{v}_1^\top + \left(\alpha - \frac{q}{p+q}\right)\mathbf{v}_2^\top\right]\mathbf{P}^t && (2.22)\\
&= \left[\left(\mathbf{P}^t\right)^\top \mathbf{v}_1 + \left(\alpha - \frac{q}{p+q}\right)\left(\mathbf{P}^t\right)^\top \mathbf{v}_2\right]^\top && (2.23)\\
&= \left[\left(\mathbf{P}^\top\right)^t \mathbf{v}_1 + \left(\alpha - \frac{q}{p+q}\right)\left(\mathbf{P}^\top\right)^t \mathbf{v}_2\right]^\top && (2.24)\\
&= \left[1^t\mathbf{v}_1 + \left(\alpha - \frac{q}{p+q}\right)(1-q-p)^t\,\mathbf{v}_2\right]^\top && (2.25)\\
\mathbf{p}(t) &= \bar{\pi} + \left(\alpha - \frac{q}{p+q}\right)(1-p-q)^t\mathbf{v}_2^\top. && (2.26)
\end{aligned}$$

Assim quando $t \to \infty$, $\mathbf{p}(t) \to \bar{\pi}$ e essa convergência será tão rápida quanto menor for o fator $|1-p-q|$. Observe também que se $\alpha = q/(p+q)$, então a cadeia já se encontra no estado estacionário em $t = 0$ e portanto $\mathbf{p}(t) = \bar{\pi}$, $\forall t \geq 0$.

Teorema 2.2. *Considere uma cadeia finita, irredutível e ergódica com matriz de transição* $\mathbf{P}$*. Se existem números não negativos* $\bar{\pi} = (\pi_0, \ldots \pi_n)$ *tal que* $\sum_{i=1}^n \pi_i = 1$ *e se* $\forall i, j \in \Omega$

$$\pi_i P_{i,j} = \pi_j P_{j,i}, \tag{2.27}$$

então $\bar{\pi}$ *é a distribuição estacionária correspondente a* $\mathbf{P}$*.*

Prova: Considere a coluna de índice j de $\bar{\pi}\mathbf{P}$. Usando a equação (2.27) temos

$$\sum_{i=1}^{n} \pi_i P_{i,j} = \sum_{i=1}^{n} \pi_j P_{j,i} = \pi_j. \tag{2.28}$$

Portanto $\bar{\pi}$ satisfaz $\bar{\pi} = \bar{\pi}\mathbf{P}$. Como $\sum_i \pi_i = 1$, e pelo teorema 2.1, segue que $\bar{\pi}$ é a única distribuição estacionária da cadeia. □

Se uma cadeia obedece a equação

$$\pi_i P_{i,j} = \pi_j P_{j,i}, \tag{2.29}$$

diz-se que a cadeia é reversível no tempo. A equação 2.29 é chamada de equação do balanço detalhado.

Considere agora uma cadeia de Markov reversível e ergódica com n estados e com matriz estocástica $\mathbf{P}$, temos a seguinte proposição que trata da convergência da cadeia para o estado estacionário:

Proposição 2.1. *Sejam* $\mathbf{p}(0)$ *a distribuição inicial da cadeia,* λ_i *o* $i-$*ésimo autovalor de* $\mathbf{P}$*,* Ψ_i^R *e* Ψ_i^L *os* $i-$*ésimos autovetores a direita e a esquerda respectivamente da matriz estocástica* $\mathbf{P}$*. Então*

$$\mathbf{P}^t = \sum_{i=1}^{n} \lambda_i^t \Psi_i^R \Psi_i^L \quad e \quad \mathbf{p}(t) = \bar{\pi} + \sum_{i=2}^{n} \lambda_i^t \langle \mathbf{p}(0), \Psi_i^R \rangle \Psi_i^L. \tag{2.30}$$

Prova: Ver apêndice B. □

O termo $\langle \mathbf{p}(0), \Psi_i^R \rangle$ é a projeção do estado inicial no autovetor Ψ_i^R e que chamaremos de $\alpha_i(0)$. Assim podemos escrever (2.30) como

$$\mathbf{p}(t) = \bar{\pi} + \sum_{i=2}^{n} \lambda_i^t \alpha_i(0) \Psi_i^L. \tag{2.31}$$

Como $1 = \lambda_1 > \lambda_2 \geq \lambda_3 \geq \cdots \geq \lambda_n \geq -1$, vamos definir $\lambda_{max} = \max\{|\lambda_i|; 2 \leq i \leq n\}$ e obter o comportamento assintótico

$$\mathbf{p}(t) = \bar{\pi} + \mathcal{O}(\lambda_{max}^t) \sum_{i=2}^{n} \alpha_i(0) \Psi_i^L. \tag{2.32}$$

Percebe-se que quando $t \to \infty$, λ_{max} é o autovalor que determina a velocidade de convergência da cadeia para o estado estacionário, pois os outros termos da soma vão a zero mais rápido que $(\lambda_{max})^t$. Assim se tivermos uma cadeia de Markov com matriz de transição P e se de alguma forma for possível estimar seu segundo autovalor, seja de forma analítica ou numérica, podemos saber de antemão a taxa de convergência da cadeia. Isso é de extrema importância, pois se estamos desenvolvemos um algoritmo cujo objetivo é transicionar pelos estados da cadeia obedecendo uma distribuição $\bar{\pi}$, o tempo de convergência para a distribuição estacionária $\bar{\pi}$ possui implicância direta no tempo de execução do algoritmo.

2.3 Caminhadas aleatórias em grafos não direcionados

Seja $G = (V, E)$ um grafo finito, não direcionado e conexo. Além disso, o conjunto dos vizinhos de i será $N(i)$ e o grau do vértice i será denotado por $d(i)$.

Definição 2.10. *Uma caminhada aleatória em* G *é uma cadeia de Markov cujo espaço de estado é* V *e*

a matriz de transição é

$$P_{i,j} = \begin{cases} 1/d(i), & \text{se } j \in N(i) \\ 0, & \text{caso contrário,} \end{cases} \tag{2.33}$$

ou seja, quando a cadeia está no vértice i, ela examina todos os vizinhos de i e escolhe um vértice vizinho com probabilidade uniforme e muda para aquele vértice.

Teorema 2.3. *Uma caminhada aleatória em G converge para a distribuição estacionária $\bar{\pi}$, onde*

$$\pi_v = \frac{d(v)}{2|E|}. \tag{2.34}$$

Prova: Primeiro vamos mostrar que π_v é uma distribuição de probabilidades. Sabe-se que $\sum_{v\in V} d(v) = 2|E|$, portanto

$$\sum_{v\in V} \pi_v = \sum_{v\in V} \frac{d(v)}{2|E|} = \frac{1}{2|E|} \sum_{v\in V} d(v) = 1. \tag{2.35}$$

Seja $\mathbf{P}$ a matriz de transição desta cadeia de Markov. A componente v da relação $\bar{\pi} = \bar{\pi}\mathbf{P}$ é

$$\pi_v = \sum_{u\in N(v)} \pi_u P_{u,v} = \sum_{u\in N(v)} \frac{d(u)}{2|E|} \frac{1}{d(u)} = \frac{|N(v)|}{2|E|} = \frac{d(v)}{2|E|}. \tag{2.36}$$

□

Exemplo 2.5. *Considere uma caminhada aleatória sobre o grafo da figura 2.3. A matriz de transição é dada por:*

$$\mathbf{P} = \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ \frac{1}{3} & 0 & \frac{1}{3} & \frac{1}{3} & 0 \\ \frac{1}{4} & \frac{1}{4} & 0 & \frac{1}{4} & \frac{1}{4} \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

e a distribuição estacionária é

$$\bar{\pi} = \begin{pmatrix} \frac{2}{12} & \frac{3}{12} & \frac{4}{12} & \frac{2}{12} & \frac{1}{12} \end{pmatrix}. \tag{2.37}$$

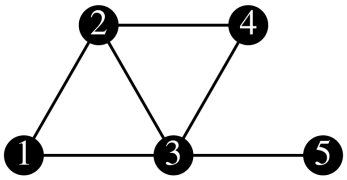

Figura 2.3: Um exemplo de grafo com 6 arestas e $V = \{1, 2, 3, 4, 5, 6\}$.

2.4 Distância total de variação

No exemplo 2.1 foi utilizada a distância Euclidiana para medir se duas distribuições estavam próximas. Pórém existem outras medidas de distância mais utilizadas quando se trata de distribuições de probabilidades. Uma delas é a distância total de variação (distância estatística) definida a seguir.

Definição 2.11. *Sejam μ_1 e μ_2 duas distribuições em Ω. A distância de variação total entre μ_1 e μ_2 é*

$$\|\mu_1 - \mu_2\|_{VT} = \frac{1}{2}\sum_{x\in\Omega} |\mu_1(x) - \mu_2(x)|. \tag{2.38}$$

Uma outra forma de escrever a distância total de variação entre duas distribuições μ_1 e μ_2 é dada pelo lema:

Lema 2.1. *Para qualquer $A \subseteq \Omega$, e $\mu_i(A) = \sum_{x\in A} \mu_i(x)$, $i = 1, 2$, temos*

$$\|\mu_1 - \mu_2\|_{VT} = \max_{A\subset\Omega} |\mu_1(A) - \mu_2(A)|. \tag{2.39}$$

Prova: Seja $S^+ \subseteq \Omega$ o conjunto de estados tais que $\mu_1(x) \geq \mu_2(x)$ e $S^- \subseteq \Omega$ o conjunto de estados tais que $\mu_2(x) \geq \mu_1(x)$, ver figura (2.4).

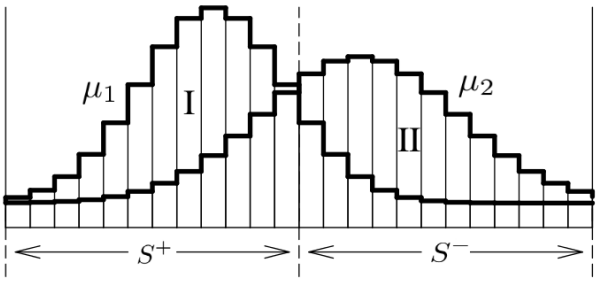

Figura 2.4: Lembrando $S^+ = \{x \in \Omega | \mu_1(x) \geq \mu_2(x)\}$ e $S^- = \{x \in \Omega | \mu_2(x) \geq \mu_1(x)\}$. A região I possui área $\mu_1(S^+) - \mu_2(S^+)$ e a região II possui área $\mu_2(S^-) - \mu_1(S^-)$. A área abaixo das curvas μ_1 e μ_2 deve ser 1, portanto a área das regiões I e II são iguais e valem $\|\mu_1 - \mu_2\|_{VT}$.

Percebe-se que

$$\max_{A\subseteq\Omega} \mu_1(A) - \mu_2(A) = \mu_1(S^+) - \mu_2(S^+) \tag{2.40}$$

e

$$\max_{A\subseteq\Omega} \mu_2(A) - \mu_1(A) = \mu_1(S^-) - \mu_2(S^-) \tag{2.41}$$

Como μ_1 e μ_2 são distribuições de probabilidade, temos que

$$1 = \sum_{x\in S} \mu_1(x) = \sum_{x\in S^+} \mu_1(x) + \sum_{x\in S^-} \mu_1(x) = \mu_1(S^+) + \mu_1(S^-) = \mu_2(S^+) + \mu_2(S^-), \tag{2.42}$$

entao

$$\mu_1(S^+) - \mu_2(S^+) = \mu_2(S^-) - \mu_1(S^-). \tag{2.43}$$

Portanto

$$\max_{A\subseteq\Omega} |\mu_1(A) - \mu_2(A)| = |\mu_1(S^+) - \mu_2(S^+)| = |\mu_2(S^-) - \mu_1(S^-)| \tag{2.44}$$

$$\begin{aligned} 2\max_{A\subseteq\Omega}|\mu_1(A)-\mu_2(A)| &= |\mu_1(S^+)-\mu_2(S^+)|+|\mu_1(S^-)-\mu_2(S^-)| \\ &= \sum_{x\in\Omega}|\mu_1(x)-\mu_2(x)| \\ &= 2\|\mu_1-\mu_2\|_{VT}. \end{aligned} \tag{2.45}$$

Onde a última igualdade foi obtida de (2.38). Finalmente

$$\|\mu_1-\mu_2\|_{VT} = \max_{A\subseteq\Omega}|\mu_1(A)-\mu_2(A)| \tag{2.46}$$

□

No caso do exemplo 2.4, usando a equação (2.26) e (2.38), a distância total de variação entre $\mathbf{p}(t)$ e $\bar{\pi}$ é

$$\|\mathbf{p}(t)-\bar{\pi}\|_{VT} = \frac{1}{2}\sum_{x\in\Omega}|\mathbf{p}(t,x)-\bar{\pi}(x)| = \frac{1}{2}\sum_{x\in\Omega}\left|\left(\alpha-\frac{q}{p+q}\right)(1-p-q)^t\mathbf{v}_2(x)\right|. \tag{2.47}$$

$$= \frac{1}{2}\left|\alpha-\frac{q}{p+q}\right||1-p-q|^t\sum_{x\in\Omega}|\mathbf{v}_2(x)| \tag{2.48}$$

$$= \left|\alpha-\frac{q}{p+q}\right||1-p-q|^t, \tag{2.49}$$

onde $\mathbf{p}(t,x)$ denota a x-ésima componente de $\mathbf{p}(t)$. A equação (2.49) mostra que a distância total de variação entre as duas distribuições decresce exponencialmente rápido a medida que t aumenta.

2.5 Tempo de mistura

Agora desejamos saber quantos passos são necessários para que a distribuição esteja distante um valor $\epsilon > 0$ do equilíbrio.

Definição 2.12. *Dado $\epsilon > 0$, o $\epsilon-$tempo de mistura τ_ϵ é o menor t tal que para qualquer distribuição inicial $\mathbf{p}(0)$, a distância até a distribuição de equilíbrio é no máximo ϵ, ou seja*

$$\tau_\epsilon = \min\{t; \max_{\mathbf{p}(0)}\|\mathbf{p}(t)-\bar{\pi}\|_{VT} \leq \epsilon\}. \tag{2.50}$$

A figura (2.5) mostra um exemplo de decaimento da distância entre as distribuições $\mathbf{p}(t)$ e $\bar{\pi}$ como função do tempo t. O tempo de mistura está indicado pelo tempo onde a distância entre as duas distribuições alcança ϵ.

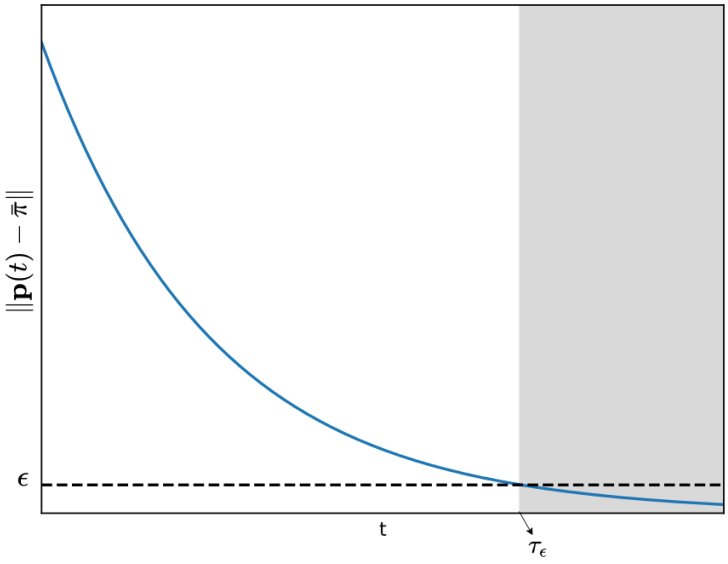

Figura 2.5: Distância $\|\mathbf{p}(t) - \bar{\pi}\|$ como função do tempo. A região em cinza mostra que a cadeia alcançou o estado estacionário (com um erro de no máximo ϵ).

Para amostrar o espaço de estados em tempo razoável, desejamos que o tempo de mistura seja polinomial no tamanho da entrada n do algoritmo. Neste caso dizemos que o tempo de mistura é polinomial.

CAPÍTULO 3

AMOSTRAGEM E CONTAGEM

Neste capítulo será mostrado como utilizar aleatoriedade para resolver problemas computacionais. No caso mais simples, utilizaremos o método de Monte Carlo para calcular o valor da constante π. No caso mais elaborado, será mostrado como unir o método de Monte Carlo e cadeias de Markov para efetuar amostragens. Serão definidos algoritmos de aproximação FPRAS e de amostragem FPAUS e por último será mostrado um caso que exemplifica a equivalência entre essas duas classes de algoritmos.

3.1 Método de Monte Carlo

Suponha que desejamos estimar numericamente o valor da constante π. A área de um círculo de raio $r = 1$ vale $\pi\, r^2 = \pi$. Assim o valor de π é

$$\pi = 4 \int_0^1 f(x)\, dx = 4 \int_0^1 \sqrt{1 - x^2}\, dx, \tag{3.1}$$

onde $f(x) = \sqrt{1 - x^2}$ representa a parte que fica no primeiro quadrante de um círculo de raio unitário, conforme mostra a Figura (3.1).

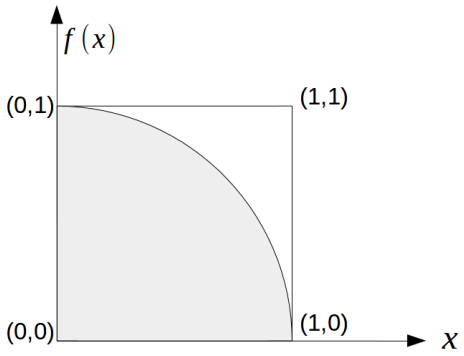

Figura 3.1: Função $f(x) = \sqrt{1 - x^2}$.

A constante π será estimada da seguinte maneira: seja (x, y) um ponto escolhido aleatoriamente, onde

x e y são escolhidos no intervalo $[0, 1]$.

Em um dado sorteio do ponto (x, y), considere $i \in [1, m]$ a variável que indexa o sorteio e m o número de sorteios. Vamos definir a variável aleatória Z_i como

$$Z_i = \begin{cases} 1, & \text{se } x^2 + y^2 \leq 1 \\ 0, & \text{caso contrário.} \end{cases} \tag{3.2}$$

Z_i conta o número de pontos que caíram abaixo da curva $f(x)$. A probabilidade de um ponto cair abaixo da curva $f(x)$ é

$$\mathsf{Pr}(Z = 1) = \frac{\pi}{4}. \tag{3.3}$$

Considere $W = \sum_{i=1}^{m} Z_i$, temos que

$$\mathbf{E}[W] = \mathbf{E}\left[\sum_{i=1}^{m} Z_i\right] = \sum_{i=1}^{m} \mathbf{E}[Z_i] = \frac{m\pi}{4}. \tag{3.4}$$

Assim se m for suficientemente grande, podemos estimar π por

$$\pi \approx W' = \frac{4}{m} W = \frac{4}{m} \sum_{i=1}^{m} Z_i. \tag{3.5}$$

Uma pergunta que surge neste ponto é quão grande deve ser m para que W' seja uma "boa" estimativa de π? De maneira mais formal, desejamos saber $\mathsf{Pr}(|W' - \pi| \geq \epsilon\pi)$ para um dado ϵ arbitrário. Para responder isso, vamos relembrar do limitante de Chernoff.

Teorema 3.1. *Sejam $X_1, X_2, \ldots, X_n$, experimentos de Poisson independentes, $X = \sum_{i=1}^{n} X_i$ e $\mu = \mathbf{E}[X]$. Para $0 < \epsilon < 1$ temos que*

$$\mathsf{Pr}(|X - \mu| \geq \epsilon\mu) \leq 2\exp(-\mu\epsilon^2/3). \tag{3.6}$$

Prova. Ver [1]. □

Aplicando este limitante ao nosso problema, obtemos

$$\begin{aligned} \mathsf{Pr}(|W' - \pi| \geq \epsilon\pi) &= \mathsf{Pr}\left(|W - \frac{m\pi}{4}| \geq \frac{\epsilon m\pi}{4}\right) \\ &= \mathsf{Pr}\left(|W - \mathbf{E}[W]| \geq m\mathbf{E}[W]\right) \\ &\leq 2\exp(-m\pi\epsilon^2/12). \end{aligned} \tag{3.7}$$

Portanto, a probabilidade da estimativa de π estar errada cai exponencialmente rápido quando aumentamos o número de amostras m.

Vamos definir de forma mais geral uma classe de algoritmos de aproximação:

Definição 3.1. *Um algoritmo aleatorizado fornece uma (ϵ, δ)-aproximação para um valor V se a saída X de um algoritmo satisfaz:*

$$\mathsf{Pr}(|X - V| \leq \epsilon V) \geq 1 - \delta. \tag{3.8}$$

No exemplo anterior, temos uma (ϵ, δ)-aproximação para π. Escolhendo m suficientemente grande, temos

$$2\exp(-m\pi\epsilon^2/12) \leq \delta, \tag{3.9}$$

ou

$$m \geq \frac{12\ln(2/\delta)}{\pi\epsilon^2}. \tag{3.10}$$

Usando novamente o limitante de Chernoff, vamos agora generalizar a ideia de (ϵ, δ)-aproximação e relacionar o número de amostras m com a qualidade da aproximação.

Teorema 3.2. *Sejam $X_1, X_2, \ldots, X_m$, variáveis aleatórias, com $\mu = \mathbf{E}[X_i]$. Se $m \geq 3\ln(2/\delta)/(\mu\epsilon^2)$, então*

$$\mathsf{Pr}\left(\left|\frac{1}{m}\sum_{i=0}^{m} X_i - \mu\right| \geq \epsilon\mu\right) \leq \delta, \tag{3.11}$$

ou seja, m amostras fornecem uma (ϵ, δ)-aproximação para μ.

Prova: Ver [1]. □

3.2 FPRAS e contagem

No caso mais geral, desejamos um algoritmo que receba como entrada uma instância x e forneça uma aproximação $V(x)$ para um problema. Por exemplo, x é um grafo G e $V(G)$ é o número de conjuntos independentes deste grafo.

Definição 3.2. *Um FPRAS (fully polynomial randomized approximation scheme) para um problema é um algoritmo aleatorizado para o qual, dado uma entrada x e quaisquer parâmetros $\epsilon > 0$ e $0 < \delta < 1$, o algoritmo tem como saída uma $(\epsilon, \delta)-$aproximação para $V(x)$ em tempo polinomial em $1/\epsilon$, $\ln(1/\delta)$ e no tamanho da entrada x.*

Vamos agora ver duas aplicações de FPRAS no problema de contagem de cláusulas na forma normal disjuntiva. Seja C_i uma cláusula conjuntiva com l_i literais e considere que existem n variáveis x_i. Uma fórmula F está na forma normal disjuntiva FND se ela é uma disjunção de cláusulas conjuntivas:

$$F = C_1 \vee C_2 \cdots \vee C_t, \tag{3.12}$$

como por exemplo

$$F = (x_1 \wedge \bar{x}_2 \wedge x_3) \vee (x_2 \wedge x_4) \vee (\bar{x}_1 \wedge x_3 \wedge x_4). \tag{3.13}$$

Considere o seguinte problema: desejamos saber de quantas maneira a fórmula F pode ser satisfeita. Este problema pertence à classe $\#P-$Completo [1]. Outros problemas que pertencem a esta classe incluem: contar o número de ciclos Hamiltonianos em um grafo e contar o número de emparelhamentos perfeitos em um grafo bipartido.

3.2.1 Monte Carlo para contagem FND - abordagem simples

Para resolver este problema, vamos usar uma abordagem semelhante àquela usada no cálculo do π. Naquele caso foram gerados pontos de maneira uniforme em um quadrado de 1×1 e contado a fração destes pontos que caem dentro do círculo e assim foi estimado o valor de π. Agora vamos gerar atribuições aleatórias para as n variáveis e ver para quantas dessas atribuições a fórmula F é satisfeita. Vamos mostrar que esta abordagem não é a mais adequada.

Seja $c(F)$ o número de formas que a fórmula F pode ser satisfeita. O algoritmo 1 apresenta uma aproximação para $c(F)$.

Algorithm 1 Algoritmo de contagem FND

Entrada: Uma fórmula F com n variáveis
Saída: $Y =$ aproximação para $c\,(F)$

1: **procedure** CONTAGEMFND
2: $X \leftarrow 0$
3: **for** $k = 1$ até m **do**
4: Gere uma atribuição aleatória para as n variáveis escolhida uniformemente dentre as 2^n possibilidades.
5: **if** $F = 1$ **then**
6: $X \leftarrow X + 1$
7: **return** $Y \leftarrow$(X/m) 2^n

Seja X_k igual a 1 se a $k-$ésima iteração do algoritmo produz $F = 1$ e X_k igual a 0 caso contrário. Portanto X_k é uma variável aleatória que recebe valor 0 ou 1. A probabilidade de X_k receber o valor 1 é $p_k = c\,(F)/2^n$. Seja $X = \sum_{k=1}^{m} X_k$, então o valor esperado para a saída Y do algoritmo é

$$\mathbf{E}[Y] = \frac{\mathbf{E}[X]2^n}{m} = \frac{2^n}{m}\sum_{k=1}^{m}\mathbf{E}[X_k] = c\,(F). \tag{3.14}$$

Usando o Teorema 3.2 e fazendo $\mu = c\,(F)/2^n$, vemos que Y fornece uma (ϵ, δ)-aproximação para $c\,(F)$ quando

$$m \geq \frac{3 \cdot 2^n \ln(2/\delta)}{\epsilon^2 c(F)}. \tag{3.15}$$

Para analisar a ordem de grandeza de m, vamos dividir em 2 casos

1. $c\,(F) \approx 2^n/\alpha(n)$, onde $\alpha(n)$ é um polinômio. Neste caso m se reduz a

$$m \geq \frac{3\alpha(n)\ln(2/\delta)}{\epsilon^2} = g\left(n, 1/\epsilon, \ln(1/\delta)\right), \tag{3.16}$$

   onde $g(n, 1/\epsilon, \ln(1/\delta))$ é uma função polinomial em n, $1/\epsilon$ e $\ln(1/\delta)$.

2. $c\,(F) \approx \alpha(n)$. Assim m é

$$m \geq g(n, 1/\epsilon, \ln(1/\delta))\, 2^n. \tag{3.17}$$

   Neste caso é necessário um número exponencial de amostras (e também um tempo exponencial de execução), proporcional a 2^n.

   Como exemplo trivial deste último caso, considere $F = x_1 \wedge \cdots \wedge x_n$ que possui 2^n possibilidades de atribuições para as variáveis, mas apenas uma solução.

Assim por exemplo se $c\,(F)$ é polinomial em n, n^2 ou n^3, será necessário um número exponencial de execuções do algoritmo para estimar $c\,(F)$ e diferenciar estes 3 casos.

O problema com essa amostragem é que conjunto de todas as atribuições que satisfazem F pode não ser denso o suficiente quando comparado com o conjunto de todas as atribuições possíveis.

Uma forma de resolver este problema é selecionar um subconjunto U (menor) do conjunto de todas as valorações possíveis. Este conjunto U pode ser escolhido seguindo algum critério ou propriedade específica. Se as amostras que satisfazem F forem densas o suficiente em U, o algoritmo pode funcionar corretamente dentro de determinada precisão e ainda executar em tempo polinomial (ver próxima subseção).

Pode-se ainda realizar uma amostragem onde coletamos mais dados em regiões do espaço amostral

com maior importância ou de acordo com determinada distribuição de probabilidade, como será visto na próxima seção (MCMC).

3.2.2 Monte Carlo para contagem FND - abordagem refinada

Para otimizar o algoritmo visto na seção anterior, vamos construir a Tabela 3.1. As linhas desta tabela indicam as t cláusulas da fórmula F. As colunas são todas as atribuições que satisfazem F. Se a cláusula C_i é satisfeita pela atribuição a_j, então existe um 1 na posição (i, j) da tabela e 0 caso contrário.

	a_1	a_2	a_3	$\dots$	a_l
C_1	0	(1)	0	$\dots$	0
C_2	(1)	0	0	$\dots$	0
C_3	0	1	0	$\dots$	(1)
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
C_t	1	0	(1)	$\dots$	1

Tabela 3.1: Listagem de todas as atribuições que satisfazem a fórmula F. Se a cláusula C_i é satisfeita pela atribuição a_j, então existe um 1 na posição (i, j) da tabela e 0 caso contrário. O primeiro 1 de cada coluna está indicado com um círculo.

Perceba que mais de uma atribuição pode satisfazer uma determinada cláusula. Por exemplo, se $C_1 = x_1 \wedge x_2$, então a atribuição $a_j = (x_1, x_2, \dots, x_n) = (1, 1, z, \dots, z')$ satisfaz C_1, independente do valor de z. De forma mais geral, se C_i possui l_i literais, então existem 2^{n-l_i} atribuições que satisfazem C_i. Além disso, para uma dada atribuição a_j, pode acontecer também de várias cláusulas serem satisfeitas.

Seja S_{C_i} o conjunto de todas as atribuições que satisfazem a cláusula i. Neste caso $|S_{C_i}|$ é o número de "uns" na linha i da tabela. Vamos definir o conjunto U, cujos elementos (i, a) fornecem o par cláusula e atribuições que satisfazem a cláusula (e portanto satisfazem F)

$$U = \{(i, a) | 1 \leq i \leq t \text{ e } a \in S_{C_i}\}. \tag{3.18}$$

Observe que $|U|$ é o número total de "uns" da Tabela 3.1. Perceba também que se tivermos a quantidade de "uns" $|S_{C_i}|$ da linha i, podemos somar sobre todas as linhas e encontrar $|U|$, ou seja

$$|U| = \sum_{i=1}^{t} |S_{C_i}|. \tag{3.19}$$

A quantidade que desejamos estimar é

$$c(F) = \left| \bigcup_{i=1}^{t} S_{C_i} \right|. \tag{3.20}$$

Como pode existir uma atribuição a_j que satisfaz mais de uma cláusula, por exemplo C_i e C_k, $i \neq k$, teremos dois elementos distintos em U: (i, a_j) e (k, a_j), mas na união acima teremos apenas um elemento a_j, então $c(F) \leq |U|$.

A fórmula (3.20) diz que se uma atribuição satisfaz mais de uma cláusula, ela deve ser contada apenas uma vez, portanto o número $c(F)$ que estamos buscando é a quantidade de "uns" circulados na Tabela 3.1, ou seja $l = c(F)$.

Para estimar $c(F)$ vamos definir o subconjunto S com tamanho $c(F)$ escolhendo o elemento $(i, a) \in U$ com o menor índice i, ou seja, S é o conjunto dos "uns" circulados na Tabela 3.1. Formalmente, S é

$$S = \{(i, a) | 1 \leq i \leq t, a \in S_{C_i}, a \notin S_{C_j} \text{ para } j < i\}. \tag{3.21}$$

Para realizar a amostragem e estimar $|S|$, vamos utilizar o método de Monte Carlo, semelhante ao exemplo da estimativa do π. Considere que o espaço de amostragem são todos os "uns" da Tabela 3.1 (conjunto U). Vamos amostrar m "uns" e contar quantos deles estão circulados, este será o valor X. Resumindo:

- $|S|$: Total de "uns" circulados na tabela..
- $|U|$: Total de "uns" na tabela.
- X: Total de "uns" circulados amostrados.
- m: total de "uns" amostrados.

Para amostrar uniformemente em U, primeiro escolhemos uma cláusula C_i com probabilidade proporcional ao número de atribuições que satisfazem C_i, ou seja, com probabilidade proporcional a $|S_{C_i}|$:

$$\mathsf{Pr}(\text{escolher } C_i) = \frac{|S_{C_i}|}{\sum_{i=1}^{t} |S_{C_i}|} = \frac{|S_{C_i}|}{|U|}. \tag{3.22}$$

Em seguida escolhemos uma atribuição aleatória que satisfaz C_i. Isso pode ser feito determinando o valor das l_i variáveis que satisfazem C_i e escolhendo aleatoriamente os $n - l_i$ literais restantes. Assim a probabilidade de escolher o par (C_i, a_j) é

$$\begin{aligned} \mathsf{Pr}\left(\text{escolher } (C_i, a_j)\right) &= \mathsf{Pr}(\text{escolher } C_i).\mathsf{Pr}(\text{escolher } a_j | C_i \text{ foi escolhida}) \\ &= \frac{|S_{C_i}|}{|U|} \frac{1}{|S_{C_i}|} \\ &= \frac{1}{|U|}, \end{aligned} \tag{3.23}$$

que é uniforme. Neste caso quando o número total m de amostras é grande, a seguinte proporção é verdadeira

$$\frac{X}{m} \approx \frac{|S|}{|U|} \tag{3.24}$$

como $c(F) = |S|$,

$$c(F) \approx \frac{X}{m} |U|. \tag{3.25}$$

Diferente da seção anterior onde o conjunto das atribuições que satisfaziam F não era denso o suficiente no conjunto total de amostragem, aqui o conjunto S é denso em U (pois t não é exponencialmente grande. Caso fosse exponencialmente grande, a fórmula F seria facilmente satisfeita e cairíamos no caso 1 na análise do final da seção anterior). A Tabela 3.1 possui $|S|t$ elementos e $|U|$ "uns", portanto

$$|S|.t \geq |U| \Rightarrow \frac{|S|}{|U|} \geq \frac{1}{t}. \tag{3.26}$$

A implementação desta amostragem é mostrada no Algoritmo 2.

Teorema 3.3. *O Algoritmo 2 fornece uma FPRAS para o problema da contagem DNF quando* $m = \lceil 3t \ln(2/\delta)/\epsilon^2 \rceil$.

Prova: Pela equação (3.26), a probabilidade de um "um" escolhido em U estar em S é pelo menos $1/t$. Considere $\epsilon > 0$, $\delta > 0$ e $m = \lceil 3t \ln(2/\delta)/\epsilon^2 \rceil$. Então m é polinomial em t, $1/\epsilon$ e $\ln(1/\delta)$ e o tempo de processamento de cada amostra é polinomial em t. Usando o Teorema 3.2, obtemos que X/m fornece uma $(\epsilon, \delta)-$aproximação para $c(F)/|U|$ e portanto Y fornece uma $(\epsilon, \delta)-$aproximação para $c(F)$. □

Algorithm 2 Algoritmo de contagem FND otimizado

Entrada: Uma fórmula F com n variáveis
Saída: $Y =$ aproximação para $c\,(F)$

1: **procedure** CONTAGEMFNDO
2: $X \leftarrow 0$
3: **for** $k = 1$ `até` m **do**
4: `Com probabilidade` $|S_{C_i}|/|U|$ `escolha a cláusula` C_i`. Em seguida gere uma atribuição aleatória` a_j `tal que` $C_i(a_j) = 1$.
5: **if** Se i for o menor índice tal que $C_i(a_j) = 1$ **then**
6: $X \leftarrow X + 1$
7: **return** $Y \leftarrow (X/m)\,|U|$

3.3 Monte Carlo via cadeias de Markov (MCMC)

Suponha que desejamos calcular o valor esperado de uma função $g(x)$, de acordo com uma distribuição de probabilidades $f(x)$

$$\mathbf{E}_{f(x)}[g(x)] = \int f(x)g(x)dx. \tag{3.27}$$

Este valor esperado pode ser aproximado amostrando valores x_s de acordo com a distribuição $f(x)$ e em seguida calculando a média sobre os valores $g(x_s)$ [11]. Neste caso uma aproximação para o valor esperado é

$$\mathbf{E}_{f(x)}[g(x)] \approx \frac{1}{N}\sum_{i=1}^{N} g(x_s), \tag{3.28}$$

onde

$$x_s \sim f(x), \tag{3.29}$$

ou seja, x é amostrado de acordo com a distribuição $f(x)$. Esta amostragem acima poderia ser feita pelo método de Monte Carlo visto anteriormente. Porém quando o número de dimensões aumenta, a amostragem começa a se tornar inviável, pois a quantidade de pontos que devem ser amostrados crescem exponencialmente com o número de dimensões [11]. Além disso, em muitos casos não conhecemos $f(x)$ exatamente, mas uma função $\tilde{f}(x)$ proporcional a $f(x)$ [6], denotaremos como $f(x) \propto \tilde{f}(x)$. Isso quer dizer que $\exists k \in \mathbb{R}$ tal que $f(x) = k\tilde{f}(x)$. A constante k pode ser por exemplo o inverso de uma constante de normalização c de $\tilde{f}(x)$ dada por

$$f(x) = \frac{1}{c}\tilde{f}(x), \qquad \text{onde} \qquad c = \int_{x\in X} \tilde{f}(x)dx \tag{3.30}$$

essa constante garante que $\int_X f(x)dx = 1$. O motivo de não conhecermos a constante de normalização acima é que em muitos casos X é exponencialmente grande, por exemplo $X \propto 2^n$, onde n é alguma variável do sistema, tornando inviável o cálculo de c. A pergunta que surge é: como calculamos o valor esperado na equação (3.27) apenas com $\tilde{f}(x)$? Veremos isso com o método MCMC.

O MCMC é uma ferramenta geral para amostrar um conjunto de dados de acordo com uma distribuição

de probabilidade específica. É um método de Monte Carlo, onde as amostras são obtidas por meio de uma cadeia de Markov. A ideia básica é definir uma cadeia de Markov ergódica cujo espaço de estados Ω é o espaço que desejamos amostrar e cuja distribuição estacionária π é distribuição de probabilidade $f(x)$. Desejamos que o tempo gasto amostrando determinadas regiões seja proporcional à probabilidade estacionária dessas regiões.

3.3.1 Algoritmo de *Metropolis-Hasting*

Lembrando que uma cadeia de Markov tem uma única distribuição estacionária se:

- A distribuição estacionária existe: considere uma cadeia em que cada transição é reversível, assim para todo par de estados x e y, a probabilidade de estar em x e transicionar para y é igual a probabilidade de estar em y e transicionar para x, ou seja $\pi_x P_{x,y} = \pi_y P_{y,x}$. Esta condição é suficiente mas não necessária para a existência da distribuição estacionária.
- A distribuição é única: isto é garantido para uma cadeia ergódica, ou seja: (i) a cadeia é aperiódica (o sistema não retorna para o mesmo estado em intervalos fixos) e (ii) a cadeia é positivo recorrente (retorna para qualquer estado em um tempo finito de transições).

Usando a equação do balanço detalhado para cadeias reversíveis temos

$$\pi_x P_{x,y} = \pi_y P_{y,x}, \tag{3.31}$$

podemos reescrever esta equação na forma

$$\frac{P_{x,y}}{P_{y,x}} = \frac{\pi_y}{\pi_x}. \tag{3.32}$$

Vamos decompor a probabilidade de transição em duas partes: (i) $h(y|x)$, que é a probabilidade condicional de propor um novo estado y, dado o estado atual x e (ii) $\alpha(x,y)$, que é a probabilidade de aceitação do novo estado y. Assim escrevemos a probabilidade de transição como

$$P_{x,y} = h(y|x)\alpha(x,y). \tag{3.33}$$

Neste caso a equação (3.32) fica

$$\frac{\alpha(x,y)}{\alpha(y,x)} = \frac{\pi_y}{\pi_x}\frac{h(x|y)}{h(y|x)}. \tag{3.34}$$

Uma escolha comum para a probabilidade de aceitação que satisfaz a equação acima é

$$\alpha(x,y) = \min\left(1, \frac{\pi_y}{\pi_x}\frac{h(x|y)}{h(y|x)}\right). \tag{3.35}$$

Para mostrar que (3.35) satisfaz (3.34), considere dois casos:

1. $\pi_y h(x|y) > \pi_x h(y|x)$. Neste caso

$$\alpha(x,y) = 1 \quad \text{e} \quad \alpha(y,x) = \frac{\pi_x}{\pi_y}\frac{h(y|x)}{h(x|y)} \tag{3.36}$$

e portanto

$$\frac{\alpha(x,y)}{\alpha(y,x)} = \frac{\pi_y}{\pi_x}\frac{h(x|y)}{h(y|x)}. \tag{3.37}$$

2. $\pi_y h(x|y) < \pi_x h(y|x)$. Neste caso

$$\alpha(x,y) = \frac{\pi_y}{\pi_x}\frac{h(x|y)}{h(y|x)} \quad \text{e} \quad \alpha(y,x) = 1 \tag{3.38}$$

e portanto

$$\frac{\alpha(x,y)}{\alpha(y,x)} = \frac{\pi_y}{\pi_x}\frac{h(x|y)}{h(y|x)}. \tag{3.39}$$

O Algoritmo de *Metropolis-Hasting* é descrito a seguir para uma variável aleatória contínua, ver [11]. Seja $f(x)$ uma função de distribuição de probabilidades (PDF - *Probability Distribution Function*) de onde desejamos amostrar e definida em um domínio D . O algoritmo *Metropolis-Hasting* é o seguinte:

1. Escolha uma PDF $h(u|x)$ definida no domínio D. A função $h(u|x)$ escolhida é chamada de função distribuição de proposta. Dado a posição atual x, o ponto u é um vizinho próximo de x e candidato para substituí-lo, sujeito à PDF $h(u|x)$. Esta PDF está condicionada a x, mas isso nem sempre é necessário.

2. Escolha um valor inicial x_i com $i = 1$, no domínio D. Este valor inicial não deve ser um valor altamente improvável. O valor inicial deve estar numa região com pontos mais prováveis de ocorrer, de acordo com $f(x)$.

3. Amostre um valor u_i de h usando algum procedimento apropriado.

4. Calcule a razão R, que é conhecida como razão de *Metropolis* ou razão de *Hasting*.

$$R(x_i, u_i) = \frac{f(u_i)}{f(x_i)}\frac{h(x_i|u_i)}{h(u_i|x_i)}. \tag{3.40}$$

5. Defina

$$\alpha(x_i, u_i) = \min\left[1, R(x_i, u_i)\right]. \tag{3.41}$$

e amostre um valor ρ de uma PDF uniforme no intervalo $[0, 1]$.

6. Aceite o valor proposto u_i com probabilidade α ou rejeite com probabilidade $1 - \alpha$, ou seja,

$$x_{i+1} = \begin{cases} u_i, & \text{se } \rho \leq \alpha \\ x_i, & \text{caso contrário.} \end{cases} \tag{3.42}$$

7. Substitua i por $i + 1$ e repita os passos de 3 a 7 até que todas as amostras desejadas tenham sido amostradas.

Resumidamente, dado um ponto x_i, é sorteado aleatoriamente um novo ponto u_i na vizinhança de x_i. O ponto x_{i+1} torna-se u_i com uma probabilidade de aceitação $\min[1, f(u_i)f(x_i)/(h(x_i|u_i)h(u_i|x_i))]$ (caso contrário, x_{i+1} fica com o valor antigo x_i). O seguinte exemplo ilustra com mais clareza como funciona este algoritmo.

Exemplo 3.1. *Metropolis-Hasting para amostrar uma função distribuição de probabilidade*

Considere que desejamos gerar amostras de acordo com a função $f(x) \propto 0.3e^{-0.2x^2} + 0.7e^{-0.2(x-10)^2}$. Observe que não é necessário saber a constante de normalização de $f(x)$, pois em $R(x_i|u_i)$ aparece a razão entre $f(u_i)$ e $f(x_i)$. Considere que a função de proposta é uma distribuição normal dada por

$h(u_i|x_i) = \mathcal{N}(x_i, 100)$. Isso significa que para cada x_i, o novo ponto proposto u_i é escolhido de acordo com uma distribuição normal com variância 100 em torno do ponto x_i. A Figura (3.2) mostra os histogramas dos pontos amostrados para $i = 100, 500, 1000$ e 5000 iterações.

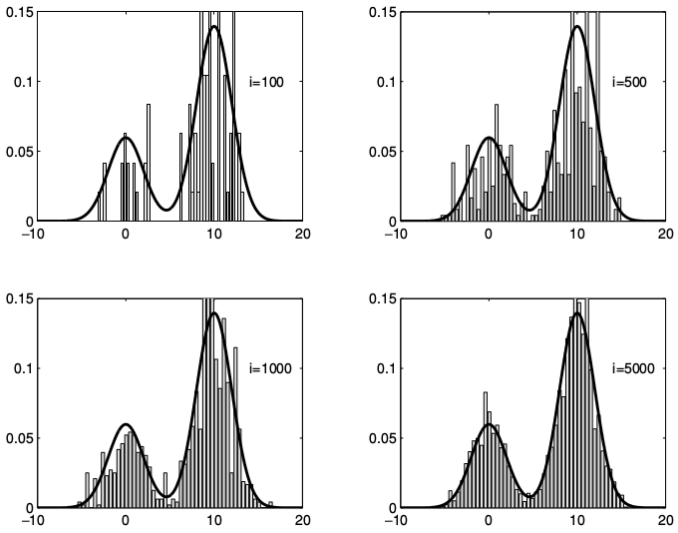

Figura 3.2: Distribuição alvo $f(x)$ e histograma das amostras obtidas por MCMC para $i = 100, 500, 1000$ e 5000 iterações. Extraído de [6].

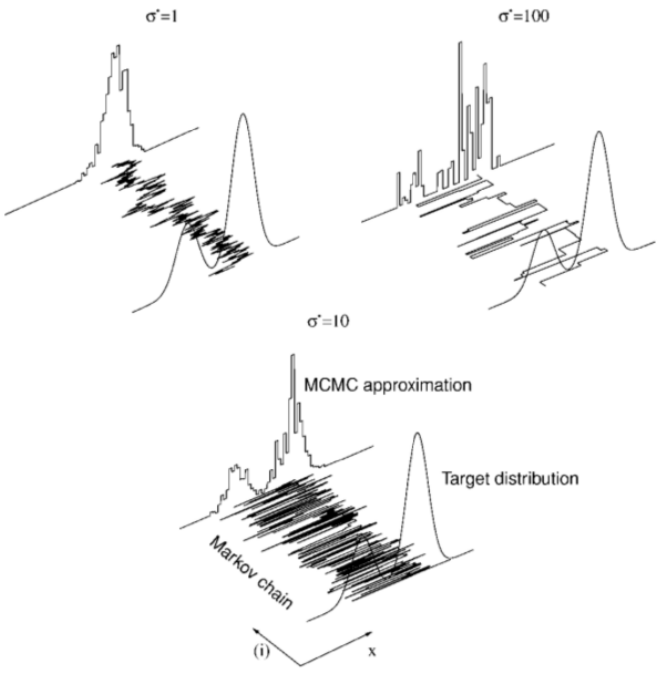

Figura 3.3: Amostragens obtidas usando o algoritmo *Metropolis-Hasting* com função de proposta Gaussiana e diferentes variâncias (tamanho do passo na caminhada aleatória). Extraído de [6].

A Figura (3.3) mostra a caminhada aleatória no espaço de amostragem utilizando diferentes tamanhos de passos (variância da função de proposta). Observe que quando o passo é muito pequeno, a caminhada aleatória não consegue caminhar por todo espaço de amostragem e o histograma dos pontos obtidos difere

da distribuição $f(x)$. Quando o passo é muito grande (variância alta), a caminhada aleatória também não consegue amostrar corretamente $f(x)$.

Um caso particular do algoritmo *Metropolis-Hasting* é o algoritmo de *Metropolis*. O algoritmo *Metropolis* realiza uma caminhada aleatória simétrica sobre os estados. Neste caso a função de proposta h é simétrica, ou seja, $h(u_i|x_i) = h(x_i|u_i)$ e a razão R torna-se

$$R(x_i, u_i) = \frac{f(u_i)}{f(x_i)}. \tag{3.43}$$

Os algoritmos *Metropolis-Hasting* e *Metropolis* geram uma cadeia de estado onde o próximo estado depende apenas do estado atual e não dos anteriores, então por isso eles são processos de Markov. Combinando isso com o fato de que as novas posições são escolhidas usando um método do tipo Monte Carlo, obtemos o nome do método Monte Carlo via Cadeias de Markov comumente conhecido pela sigla em inglês MCMC.

3.4 Amostragem de conjuntos independentes

Vamos agora aplicar o método MCMC em um caso discreto: amostragem de conjuntos independentes em um grafo $G = (V, E)$.

Seja $x \in \Omega$ um estado da cadeia. O conjunto de estados alcançáveis a partir de x em uma transição é representado pelos vizinhos de x, denotado por $N(x)$. Além disso, vamos adotar a restrição $y \in N(x) \Leftrightarrow x \in N(y)$. Não confundir $N(x)$ com os vizinhos de um vértice $v \in V$. O espaço de estados Ω da cadeia são todos os conjuntos independentes do grafo G, ou seja, se I_x é um conjunto independente no grafo G, entao I_x é um estado em Ω. Para definir as transições, poderíamos fazer uma caminhada aleatória no espaço de estados Ω, mas a distribuição de probabilidade π_x estacionária de uma caminhada aleatória em um grafo é proporcional ao grau do vértice x. O que desejamos na realidade é que a distribuição de probabilidade estacionária tenha uma forma qualquer $\pi_x = b(x)/B$, onde $\forall x \in \Omega$, $b(x) > 0$ e $B = \sum_{x\in\Omega} b(x)$ (fator de normalização) deve ser finito.

Como visto no algoritmo de *Metropolis-Hasting*, não é necessário conhecer o fator de normalização, pois o algoritmo depende apenas das razões da função de distribuição para dois pontos diferentes, portanto não é necessário conhecer o valor de B. As transições de estados é dada pelo seguinte lema:

Lema 3.1. *Para um espaço finito Ω e uma vizinhança $\mathbf{N} = \{N(x)|x \in \Omega\}$, seja $\tilde{N} = \max|N(x)|$ e seja M qualquer número tal que $M \geq \tilde{N}$. Para todo $x \in \Omega$, seja $\pi_x > 0$ a distribuição desejada no estado estacionário. Considere uma cadeia de Markov onde:*

$$P_{x,y} = \begin{cases} (1/M)\min[1, \pi_y/\pi_x] & \text{se } x \neq y \text{ e } y \in N(x), \\ 0 & \text{se } x \neq y \text{ e } y \notin N(x), \\ 1 - \sum_{x\neq y} P_{x,y} & \text{se } x = y. \end{cases} \tag{3.44}$$

Se esta cadeia é irredutível e aperiódica, então π_x é a distribuição estacionária para x.

Prova: Vamos mostrar que a cadeia é reversível, ou seja (Teorema 2.2) $\pi_i P_{i,j} = \pi_j P_{j,i}$ e concluir que π_x é a distribuição estacionária.

Para qualquer $x \neq y$, se $\pi_x \leq \pi_y$, então $P_{x,y} = 1/M$ e $P_{y,x} = (1/M)\pi_x/\pi_y$. Neste caso temos que $\pi_x P_{x,y} = \pi_y P_{y,x}$. Por outro lado se $\pi_x > \pi_y$, então $P_{x,y} = (1/M)\pi_y/\pi_x$ e $P_{y,x} = 1/M$ e portanto $\pi_x P_{x,y} = \pi_y P_{y,x}$. Assim segue pelo Teorema 2.2 que π_x é uma distribuição estacionária. □

Exemplo 3.2. *Amostragem com probabilidade proporcional ao tamanho do conjunto independente*

Como um exemplo de aplicação, suponha que desejamos criar uma cadeia de Markov, onde na distribuição estacionária, cada conjunto independente I_x seja amostrado com uma probabilidade proporcional a $\lambda^{|I_x|}$, onde $\lambda > 0$ é uma constante, ou seja, $\pi_x = \lambda^{|I_x|}/B$, onde $B = \sum_{x' \in \Omega} \lambda^{|I_{x'}|}$. Quando $\lambda = 1$ a distribuição é uniforme. Se $\lambda > 1$, conjuntos independentes maiores têm maior probabilidade de serem amostrados que conjuntos independentes menores. Para $\lambda < 1$, os conjuntos independentes menores tem maior probabilidade de serem amostrados que conjuntos independentes maiores.

Considere a cadeia de Markov, cujos estados são conjuntos independentes do grafo $G = (V, E)$. Temos o seguinte procedimento

1. X_0 é um conjunto independente em G.
2. Para calcular X_{i+1}:
   (a) Escolha um vértice v aleatoriamente em V.
   (b) Se $v \in X_i$, faça $X_{i+1} = X_i - \{v\}$ com probabilidade $\min(1, 1/\lambda)$.
   (c) Se $v \notin X_i$ e se adicionando v a X_i ainda fornece um conjunto independente, então faça $X_{i+1} = X_i \cup \{v\}$ com probabilidade $\min(1, \lambda)$.
   (d) Caso contrário, faça $X_{i+1} = X_i$.

Este procedimento tem basicamente duas etapas, primeiro é escolhido um vértice uniformemente com probabilidade $1/M = 1/|V|$. A seguir, considerando x como estado atual e y como estado proposto, o vértice é aceito com probabilidade $\min[1, \pi_y/\pi_x]$, pois se estamos adicionando um vértice, $|I_y| = |I_x| + 1$ e $\pi_y/\pi_x = \lambda^{|I_y|-|I_x|} = \lambda$, que concorda com a etapa $2(c)$. Se estamos removendo um vértice, $|I_y| = |I_x| - 1$ e $\pi_y/\pi_x = \lambda^{|I_y|-|I_x|} = \lambda^{-1}$, que concorda com a etapa 2(b). Considerando estas duas etapas, de selecionar um vértice com probabilidade $1/M$ e aceita-lo com probabilidade $\min[1, \pi_y/\pi_x]$, temos

$$P_{x,y} = \frac{1}{M} \min(1, \pi_y/\pi_x), \tag{3.45}$$

que concorda com o Lema 3.1 acima.

3.5 FPAUS e amostragem

Definição 3.3. *Seja w a saída (aleatória) de um algoritmo de amostragem para um espaço de amostragem finito Ω. O algoritmo de amostragem gera uma amostra $\epsilon-$uniforme de Ω se, para qualquer subconjunto S de Ω,*

$$\left| \mathsf{Pr}(w \in S) - \frac{|S|}{|\Omega|} \right| \leq \epsilon. \tag{3.46}$$

Um algoritmo de amostragem é FPAUS (fully polynomial almost uniform sampler) para um problema se, para uma entrada x e um parâmetro $\epsilon > 0$, ele gera uma amostra $\epsilon-$uniforme de $\Omega(x)$ e executa em tempo que é polinomial no tamanho da entrada e $\ln(\epsilon^{-1})$.

A definição acima pode ser vista em termos da distância total de variação como: um algoritmo de amostragem retorna uma amostra $\epsilon-$uniforme se e somente a distribuição de saída D e uma distribuição uniforme U em Ω diferem por uma distância de no máximo ϵ [1], ou seja

$$\|D - U\| \leq \epsilon. \tag{3.47}$$

Vamos agora ver um caso exemplificando que é possível construir um algoritmo de contagem baseado em um algoritmo de amostragem, caso ele exista.

Exemplo 3.3. *FPRAS para contar o número de conjuntos independentes*

O objetivo aqui é mostrar que dado um FPAUS para conjuntos independentes, podemos construir um FPRAS para contar o número de conjuntos independentes. Considere G um grafo com m arestas e seja $e_1, e_2, \ldots, e_m$ uma ordenação arbitrária das arestas. Seja E_i o conjunto das primeiras i arestas de E e $G_i = (V, E_i)$. Note que $G_m = G$ e G_{i-1} é obtido de G_i removendo uma aresta.

Seja $\Omega(G_i)$ o conjunto dos conjuntos independentes em G_i. Podemos expressar o número de conjuntos independentes em G como

$$|\Omega(G)| = \frac{|\Omega(G_m)|}{|\Omega(G_{m-1})|} \times \frac{|\Omega(G_{m-1})|}{|\Omega(G_{m-2})|} \times \cdots \times \frac{|\Omega(G_1)|}{|\Omega(G_0|} \times |\Omega(G_0)|. \tag{3.48}$$

Como G_0 não possui arestas e cada vértice é um conjunto independente, então $|\Omega(G_0)| = 2^n$. Para estimar $|\Omega(G)|$ precisamos estimar as razões

$$r_i = \frac{|\Omega(G_i)|}{|\Omega(G_{i-1})|}. \tag{3.49}$$

Seja $\tilde{r}_i$ e estimativa de r_i. A estimativa de $|\Omega(G)|$ é

$$2^n \prod_{i=1}^{m} \tilde{r}_i, \tag{3.50}$$

enquanto que o valor real é

$$|\Omega(G)| = 2^n \prod_{i=1}^{m} r_i. \tag{3.51}$$

Para estimar corretamente a quantidade acima, devemos limitar o erro na quantidade

$$R = \prod_{i=1}^{m} \frac{\tilde{r}_i}{r_i}. \tag{3.52}$$

Para obter uma $(\epsilon, \delta)-$aproximação, vamos limitar o erro por meio de $\mathsf{Pr}(|R-1| \leq \epsilon) \leq 1-\delta$ e para isso usaremos o seguinte Lema

Lema 3.2. *Suponha que $\forall i, 1 \leq i \leq m$, $\tilde{r}_i$ é uma $(\epsilon, \delta)-$aproximação para r_i, então*

$$\mathsf{Pr}(|R-1| \leq \epsilon) \leq 1-\delta \tag{3.53}$$

Prova: Ver [1]. □

Portanto tudo o que precisamos é um método para obter uma $(\epsilon, \delta)-$aproximação para r_i. Isso é obtido com o método de Monte Carlo que usa FPAUS para amostrar conjuntos independentes em G (Exemplo 3.2 com $\lambda = 1$). Para estimar r_i, amostramos conjuntos independentes em G_{i-1} e calculamos a fração deles que são também conjuntos independentes em G_i, como mostrado no Algoritmo (3).

Algorithm 3 Estimativa de r_i

Entrada: Grafos $G_{i-1} = (V, E_{i-1})$ e $G_i = (V, E_i)$
Saída: $\tilde{r}_i \approx r_i$

1: **procedure** ESTIMA r_i
2: $X \leftarrow 0$
3: **for** $k = 1$ `até` $M = \lceil 1296 m^2 \epsilon^{-2} \ln(2m/\delta) \rceil$ **do**
4: `Gere uma amostra` x $(\epsilon/6m)-$`uniforme de` $\Omega(G_{i-1})$.
5: **if** $x \in \Omega(G_i)$ **then**
6: $X \leftarrow X + 1$
7: **return** $\tilde{r}_i \leftarrow$X/M

Lema 3.3. *Para $m \geq 1$ e $0 < \epsilon \leq 1$, o procedimento para estimar r_i fornece uma $(\epsilon/2m, \delta/m)-$aproximação para r_i*

Prova: Ver [1]. □

O número de amostras M é polinomial em m, ϵ e $\ln(\delta^{-1})$ e o tempo para cada amostra é polinomial no tamanho do grafo e em $\ln(\epsilon^{-1})$, portanto temos o seguinte teorema.

Teorema 3.4. *Dado um FPAUS para conjuntos independentes em qualquer grafo, podemos construir um FPRAS para o número de conjuntos independentes em um grafo G.*

Para uma teoria mais detalhada da redução de um FPAUS para FPRAS, ver [12].

CAPÍTULO 4

TÉCNICAS PARA LIMITAR O TEMPO DE MISTURA

Cadeias de Markov possuem diversas aplicações em diversas áreas e muitas vezes os algoritmos são até simples de serem desenvolvidos. O problema é que em muitos casos, não sabemos qual é o tempo de mistura da cadeia. Assim, este capítulo tem o objetivo de mostrar duas técnicas que são normalmente utilizadas para limitar o tempo de mistura. Na prática, muitas vezes é complicado de aplicar estas técnicas e provar que a cadeia possui um tempo de mistura de acordo com o desejado. Mesmo assim, estas técnicas são tradicionais e servirão para fortalecer a base e desenvolvermos mais a nossa intuição a respeito de cadeias de Markov.

4.1 Acoplamento de cadeias de Markov

Acoplamento de cadeias de Markov é utilizado para determinar a convergência da cadeia. A ideia é criar duas cópias da cadeia de Markov que iniciam em estados diferentes $X_0 = x$ e $Y_0 = y$ e evoluem em paralelo. O acoplamento é uma forma inteligente de fazer as duas cadeias convergirem uma para a outra. Com isso podemos criar um limitante para a taxa de convergência da cadeia e assim será possível saber quando é seguro amostrar dados que seguem a distribuição estacionária.

Uma propriedade do acoplamento é que se as duas cópias alcançaram o mesmo estado, então elas permanecerão transicionando para os mesmo estados em todos os tempos posteriores [13]. Além disso, se em um determinado tempo as duas cadeias alcançaram o mesmo estado, então podemos concluir que elas "esqueceram" o estado inicial. Se isso é verdade para todos os estados iniciais x e y, então temos certeza que o estado inicial foi realmente "esquecido" e portanto os estado X_t são realmente aleatórios.

Um acoplamento de uma cadeia de Markov $(\Omega, \mathbf{P})$ é uma cadeia de Markov com estados $Z_t = (X_t, Y_t)$ no espaço $\Omega \times \Omega$ tal que

$$\mathsf{Pr}(X_{t+1} = x' | Z_t = (x, y)) = \mathsf{Pr}(X_{t+1} = x' | X_t = x) = P_{x,x'} \tag{4.1}$$

$$\mathsf{Pr}(Y_{t+1} = y' | Z_t = (x, y)) = \mathsf{Pr}(Y_{t+1} = y' | Y_t = y) = P_{y,y'}, \tag{4.2}$$

ou seja, X_t e Y_t são cadeias em $(\Omega, \mathbf{P})$. Sejam $p_x(x)$, $p_y(y)$ e $q_z(x,y)$ as distribuições das variáveis X, Y e (X,Y) respectivamente. O acoplamento destas duas cadeias (cujos estados no tempo t são variáveis aleatórias) é uma distribuição conjunta $q_z(x,y)$ que possui $p_x(x)$ e $p_y(y)$ como distribuições marginais, ou seja,

$$p_x(x) = \sum_y q_z(x,y) \quad \text{e} \quad p_y(y) = \sum_x q_z(x,y). \tag{4.3}$$

Poderíamos fazer $q_z(x,y) = p_x(x)p_y(y)$ que corresponderia a duas cadeias evoluindo de forma independente no tempo. Mas este não é o único acoplamento possível. Na verdade podemos escolher qualquer $q_z(x,y)$, desde que mantenha $p_x(x)$ e $p_y(y)$ como distribuições marginais. O que será feito é escolher um acoplamento que limite a distância total de variação entre as duas distribuições.

Lema do Acoplamento 4.1. *Seja $Z_t = (X_t, Y_t)$ um acoplamento de uma cadeia de markov em um espaço de estados $\Omega \times \Omega$. Suponha que existe um tempo T tal que para todo x, $y \in \Omega$,*

$$\mathsf{Pr}(X_T \neq Y_T | X_0 = x, Y_0 = y) \leq \epsilon, \tag{4.4}$$

então

$$\tau_\epsilon \leq T. \tag{4.5}$$

Isto quer dizer que, para qualquer estado inicial, a distância de varição entre a distribuição do estado da cadeia após T passos e a distribuição estacionária é no máximo ϵ.

Prova: Considere um tempo T, $0 \leq \epsilon \leq 1$ e $A \subset \Omega$:

$$\mathsf{Pr}(X_T \in A) = \mathsf{Pr}(X_T \in A \wedge Y_T \in A) + \mathsf{Pr}(X_T \in A \wedge Y_T \notin A) \tag{4.6}$$

e

$$\mathsf{Pr}(Y_T \in A) = \mathsf{Pr}(X_T \in A \wedge Y_T \in A) + \mathsf{Pr}(X_T \notin A \wedge Y_T \in A). \tag{4.7}$$

Portanto

$$\begin{aligned} \mathsf{Pr}(X_T \in A) - \mathsf{Pr}(Y_T \in A) &= \mathsf{Pr}(X_T \in A \wedge Y_T \notin A) - \mathsf{Pr}(X_T \notin A \wedge Y_T \in A) && (4.8)\\ &\leq \mathsf{Pr}(X_T \in A \wedge Y_T \notin A) && (4.9)\\ &= \mathsf{Pr}(X_T \neq Y_T) && (4.10) \end{aligned}$$

usando a equação (4.4) e considerando $Y = \pi$, obtemos

$$\mathsf{Pr}(X_T \in A) - \pi(A) \leq \epsilon. \tag{4.11}$$

Como A é arbitrário, podemos fazer a substituição $A \to \Omega - A$ e obter

$$\begin{aligned} \mathsf{Pr}(X_T \in \Omega - A) - \pi(\Omega - A) &\leq \epsilon && (4.12)\\ 1 - \mathsf{Pr}(X_T \in A) - 1 + \pi(A) &\leq \epsilon && (4.13) \end{aligned}$$

ou

$$\mathsf{Pr}(X_T \in A) - \pi(A) \geq -\epsilon \tag{4.14}$$

e portanto

$$|\mathsf{Pr}(X_T \in A) - \pi(A)| \leq \epsilon. \tag{4.15}$$

A equação acima é válida para qualquer $X_0 = x$ e A, então

$$\max_{x,A} |\mathsf{Pr}(X_T \in A) - \pi(A)| = \max_{x} \|p_x(A) - \pi(A)\|_{VT} \leq \epsilon. \tag{4.16}$$

De acordo com a definição de tempo de mistura, $T \geq \tau_\epsilon$. □

Exemplo 4.1. *Caminhada aleatória em um hipercubo.*

Para exemplificar o acoplamento, vamos primeiro analisar a cadeia de Markov de uma caminhada aleatória em um hipercubo $n-$dimensional, ver Figura (4.1). Um estado da cadeia em um tempo t é uma coleção $X_t = (x_1, \ldots, x_n) \in \{0,1\}^n$. A caminhada aleatória será feita de maneira "preguiçosa", ou seja, com probabilidade $1/2$ ficamos no vértice atual e com probabilidade $1/2$ caminhamos para um vértice vizinho. Lembrando que dois vértices estão ligados por uma aresta se seus estados (suas coordenadas) diferem em apenas 1 posição na coleção. Para encontrar o estado do tempo $t+1$, escolhermos uma posição i na coleção de maneira aleatória dentre $\{1, \ldots n\}$ e escolhemos seu próximo valor z dentre $\{0,1\}$, também de maneira aleatória. Em seguida fazemos $x_i = z$, ou seja, $X_{t+1} = (x_1, \ldots, x_i = z, \ldots x_n)$.

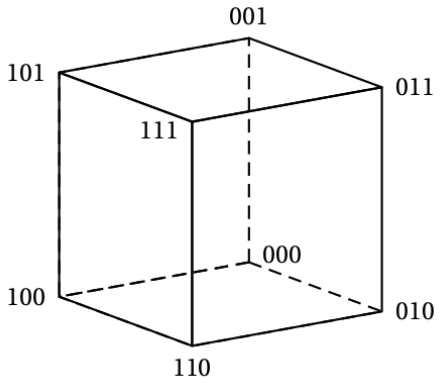

Figura 4.1: Hipercubo de dimensão 3. Observe que dois vértices são vizinhos se suas coordenadas diferem em apenas 1 bit.

Para fazer o acoplamento, teremos agora em um tempo t duas cadeias com estados $X_t = (x_1, \ldots x_n)$ e $Y_t = (y_1, \ldots y_n)$ que evoluem no tempo de forma dependente. A dependência entre elas será mantida da seguinte forma: o bit i e o seu novo valor z será o mesmo para as duas cadeias, ou seja, fazemos $X_{t+1} = \{x_1, \ldots, x_i = z, \ldots x_n\}$ e $Y_{t+1} = \{y_1, \ldots, y_i = z, \ldots y_n\}$. Após este passo, as duas cadeias concordarão no bit i para todos os tempos posteriores. Assim teremos o acoplamento das duas cadeias após os n bits terem sidos escolhidos.

Quanto tempo leva para ocorrer o acoplamento? A solução se resume ao problema do colecionador de cupons: se a cada dia recebemos um cupom e existem n tipos de cupons, quantos dias leva para coletar pelo menos 1 cupom de cada tipo?

A probabilidade de não encontrar um cupom específico em um determinado dia é $(1 - 1/n)$. Assim a probabilidade de não encontrar um determinado cupom após t dias é $(1-1/n)^t$. Como existem n cupons, o valor esperado do número de cupons faltantes após t dias é $n(1 - 1/n)^t$.

Pela desigualdade de Markov, para uma variável aleatória Z, temos que

$$\mathsf{Pr}(Z \geq a) \leq \frac{\mathbf{E}[Z]}{a}. \tag{4.17}$$

Considere que Z representa o número de cupons que faltam ser coletados:

$$\mathsf{Pr}(Z > 0) = \mathsf{Pr}(Z \geq 1) \leq \mathbf{E}[Z] = n\left(1 - \frac{1}{n}\right)^t. \tag{4.18}$$

Considerando $x = 1/n$, $n > 1$ e usando a expansão em séries de potência para $\ln(1-x)$ temos

$$\ln(1-x) = -x - \frac{x^2}{2} - \frac{x^3}{3} - \cdots \leq -x, \tag{4.19}$$

ou

$$1 - x \leq e^{-x}. \tag{4.20}$$

Portanto

$$\mathsf{Pr}(Z > 0) \leq ne^{-t/n}. \tag{4.21}$$

Escolhendo um t tal que $ne^{-t/n} = \epsilon$ e usando o Lema 4.1, encontramos

$$\tau_\epsilon \leq n\ln(n\epsilon^{-1}) = \mathcal{O}(n\ln(n)) \tag{4.22}$$

para ϵ constante. Portanto esta cadeia de Markov tem um rápido tempo de mistura.

4.2 Condutância

Fluxo de probabilidade em cadeias de Markov é muito semelhante a fluxo de eletricidade numa rede elétrica ou fluxo de água numa rede encanada. Se a rede é muito bem conectada, então rapidamente ocorre fluxo de probabilidade para todos os estados da cadeia e o estado estacionário é alcançado. Por outro lado, se existem muitos gargalos na rede, então existirão regiões com fluxo de probabilidade muito baixa e pode acontecer de levar muito tempo para a distribuição estacionária ser alcançada. A seguir será formalizado o conceito de condutância e sua relação com o tempo de mistura da cadeia de Markov.

Considere dois estados $i, j \in \Omega$ de uma cadeia de Markov. Podemos definir um fluxo de probabilidade $Q_{i,j}$ de i para j no estado estacionário como

$$Q_{i,j} = \pi_i P_{i,j}. \tag{4.23}$$

Se $Q_{i,j} = Q_{j,i}$, a cadeia é reversível como visto em seções anteriores.

Vamos agora criar uma partição $(S, \bar{S}) = (S, \Omega - S)$ e definir o fluxo de S para $\bar{S}$

$$Q(S, \bar{S}) = \sum_{i \in S, j \in \bar{S}} \pi_i P_{i,j}. \tag{4.24}$$

A probabilidade de ir em um passo de S para $\bar{S}$ dado que a cadeia está em S é

$$\Phi(S) = \frac{Q(S, \bar{S})}{\pi(S)}, \tag{4.25}$$

onde $\pi(S) = \sum_{i \in S} \pi_i$. Pensando de forma mais intuitiva, um $\Phi(S)$ baixo indica que a cadeia vai ficar muito tempo transicionando pelos estados de S e raramente vai transicionar para $\bar{S}$. De forma aproximada, a cadeia levaria um tempo proporcional a $\Phi(S)^{-1}$ para sair de S. Por outro lado se $\Phi(S)$ é alto, quer dizer que é muito fácil transicionar entre os estados da cadeia, indicando que ela é muito bem

conectada e sem gargalos. Neste caso a cadeia tem um rápido tempo de mistura.

Para definir a condutância de uma cadeia de Markov, vamos pegar o pior caso de $\Phi(S)$, ou seja, vamos escolher $\Phi(S)$ tal que S seja o conjunto mais difícil de escapar. Mas isso não basta, pois caso escolhêssemos um conjunto S com tamanho próximo de Ω e que $\pi(S) \approx 1$, dificilmente ocorreria uma transição para fora de S. Portanto vamos colocar um limite superior de $\pi(S)$ em $1/2$. Assim a condutância do grafo G é

$$\Phi = \min_{S:0<\pi(S)\leq 1/2} \Phi(S). \tag{4.26}$$

Considere agora um grafo não direcionado. Uma definição alternativa para condutância que independe dos conceitos de cadeia de Markov é

$$\Phi'(S) = \frac{|E(S,\bar{S})|}{\min(\mathrm{Vol}(S), \mathrm{Vol}(\bar{S}))}, \tag{4.27}$$

onde $E(S,\bar{S}) = \{\{i,j\}|i \in S, j \in \bar{S}\}$ e $\mathrm{Vol}(S) = \sum_{i\in S} d_i$. Assim temos a seguinte definição equivalente para Φ

$$\Phi = \min_S \Phi'(S). \tag{4.28}$$

Para verificar a equivalência, considere $\pi_i = d_i/(2|E|)$, $P_{i,j} = 1/d_i$ se $i \sim j$ e $P_{i,j} = 0$ caso contrário. Assim temos

$$\Phi = \min_{S:0<\pi(S)\leq 1/2} \frac{\sum_{i\in S, j\in\bar{S}} \pi_i P_{i,j}}{\sum_{i\in S}\pi_i} = \min_{S:0<\pi(S)\leq 1/2} \frac{\sum_{i,j,i\sim j} d_i/(d_i 2|E|)}{\sum_{i\in S} d_i/(2|E|)} \tag{4.29}$$

$$= \min_{S:0<\mathrm{Vol(S)}\leq(1/2)\mathrm{Vol}(V)} \frac{\sum_{i,j,i\sim j} 1}{\mathrm{Vol}(S)} = \min_S \frac{|E(S,\bar{S})|}{\min(\mathrm{Vol}(S),\mathrm{Vol}(\bar{S}))} = \min_S \Phi'(S). \tag{4.30}$$

4.2.1 Desigualdade de Cheeger

Encontrar a condutância de um grafo é um problema NP-completo [21]. Mas encontrar o segundo autovalor de uma matriz possui complexidade polinomial. Assim, relacionar estas duas grandezas pode ser de muita utilidade quando desejamos provar limitantes para o tempo de mistura e outras propriedades.

Vamos a seguir mostrar 2 definições referentes ao laplaciano de um grafo e em seguida relacionar o autovalor do laplaciano com a condutância de um grafo. Este resultado foi inicialmente estabelecido por Jeff Cheeger em 1970 [20] para variedades (espaços topológicos) e posteriormente estendido para grafos.

Definição 4.1 (Laplaciano de um grafo)**.** *Seja $G = (V, E)$ um grafo não direcionado, $\mathbf{A}$ a sua matriz de adjacência e $\mathbf{D}$ uma matriz diagonal onde a $i-$ésima diagonal é o grau do vértice i, ou seja $\mathbf{D} = diag(d_1, \ldots, d_n)$. O laplaciano de G é definido como $\mathbf{L} = \mathbf{D} - \mathbf{A}$.*

Definição 4.2 (Laplaciano normalizado)**.** *O laplaciano normalizado de G é definido por $\mathcal{L} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$ $= \mathbb{1} - \tilde{\mathbf{A}}$, onde $\tilde{\mathbf{A}} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ é a matriz de adjacência normalizada.*

Já é um fato bastante conhecido na área de teoria espectral de grafos que o segundo menor autovalor da matriz $\mathbf{L}$ é nulo se e somente o grafo G é desconexo (ver prova no apêndice C). De forma semelhante é possível provar o mesmo resultado para o segundo menor autovalor (que chamaremos de μ_2) de $\mathcal{L}$, ou seja, μ_2 é nulo se e somente o grafo G é desconexo. Assim a pergunta que surge é: considerando que μ_2 não é zero mas aproximadamente zero, podemos afirmar algo sobre a conectividade do grafo? A resposta é sim e existe um limitante, conhecido como desigualdade de Cheeger que relaciona a condutância Φ do grafo com o autovalor μ_2.

Teorema 4.1. *(Desigualdade de Cheeger) Seja μ_2 o segundo menor autovalor de $\mathcal{L}$, então*

$$\frac{\mu_2}{2} \leq \Phi \leq \sqrt{2\mu_2} \tag{4.31}$$

ou de forma equivalente

$$\frac{\Phi^2}{2} \leq \mu_2 \leq 2\Phi. \tag{4.32}$$

Prova: A prova do lado esquerdo da equação (4.31) pode ser vista no apêndice D. O restante da prova pode ser vista em [14]. □

Para algumas cadeias é possível determinar a condutância analiticamente. Também foi visto no Capítulo 2 que a taxa de convergência de uma cadeia depende basicamente do segundo maior autovalor λ_{max} da matriz estocástica, mas λ_{max} está relacionado com μ_2 ($\mu_2 = 1 - \lambda_{max}$). Então com o teorema acima, é possível investigar a taxa de convergência da cadeia de Markov analisando a estrutura do grafo de estados da cadeia.

CAPÍTULO 5

ALGORITMO *METROPOLIS* PARA O CÁLCULO DA CENTRALIDADE DE INTERMEDIAÇÃO

Medidas de centralidade são de extrema importância em redes complexas, pois permitem quantificar a importância de um vértice em uma rede. Como exemplos de aplicação, podemos determinar qual é a pessoa que exerce mais influência em uma rede social, ou quais são as pessoas mais mais propensas a espalhar uma doença (como COVID-19) em uma rede, ou ainda quais são os nós mais importantes em uma rede de internet ou uma rede de transporte.

Existem diversas medidas de centralidade. A primeira medida de centralidade a ser definida e que também é a mais simples e intuitiva de entender é o grau de um nó da rede. Em uma rede social por exemplo, é razoável pensar que a pessoa com mais amigos ou mais contatos é a que tem mais acesso a informações e possui maior influência sobre os outros. Já em uma rede de citações de artigos por exemplo, o número de citações que um artigo recebe, que é o número de arcos que apontam para o nó na rede, fornece uma medida quantitativa da importância do artigo.

Por outro lado a centralidade de grau pode fornecer uma estimativa totalmente errada da importância de um nó na rede. Considere a Figura (5.1) que mostra uma rede complexa com dois grupos, ligados por uma única "ponte", constituída de 2 arestas e um único vértice A. Se o vértice A for removido, a rede se torna desconexa, portanto A possui alta importância na rede de acordo com a medida de centralidade de intermediação (que será definida logo a frente), mas possui baixa centralidade de grau, pois o grau de A é 2.

Outras medidas de centralidade muito usadas são centralidade de *Katz*, *PageRank* e outras. Para mais detalhes do cálculo de cada uma delas, ver [15] e para uma análise da utilidade de cada uma, ver [16] que calcula a correlação entre 17 tipos de centralidades aplicadas em diferentes tipos de redes complexas.

No restante deste capítulo será definida a centralidade intermediação e o uso do algoritmo de *Metropolis* para seu cálculo.

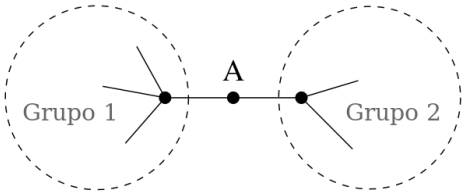

Figura 5.1: Exemplo de rede complexa onde um nó A possui baixa centralidade de grau, mas alta intermediação. Isso ocorre porque qualquer caminho do grupo 1 para o grupo 2 deve passar por A. Extraído de [15].

5.1 Centralidade de Intermediação $C(i)$

A proposta de centralidade de intermediação é atribuída a Linton Freeman [17] em 1977 no uso de redes sociais, embora alguns anos antes já existiam relatórios técnicos que utilizavam esta medida. A intermediação mede o número de vezes que um nó serve de ponte no menor caminho entre um par de vértices da rede. Em alguns casos, pode existir mais de um menor caminho entre dois vértices, como mostrado na figura (5.2).

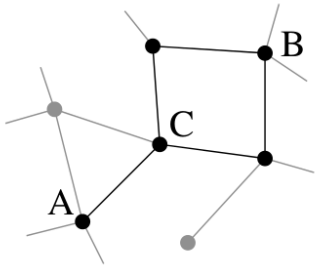

Figura 5.2: Sobreposição de 2 caminhos de A para B que passam por C.

Seja G um grafo não direcionado, sem pesos e com conjuntos V e E para vértices e arestas respectivamente, onde $|V| = n$ e $|E| = m$. Matematicamente, a intermediação do vértice i é definida da seguinte maneira: Seja $\sigma_{st}(i)$ o número de menores caminhos de s (*source*) para t (*target*) que passam por i e seja σ_{st} o número total de menores caminhos de s para t. A intermediação do vértice i é dada por

$$C(i) = \sum_{s,t} \frac{\sigma_{st}(i)}{\sigma_{st}}. \tag{5.1}$$

Muitas vezes é mais conveniente trabalhar com valores normalizados de forma que $C(i)$ fique entre 0 e 1. Para isso existem diferentes definições para a constante de normalização de $C(i)$. Neste trabalho usaremos $n(n-1)$ como constante de normalização, onde n é o número de vértices do grafo. Assim $C(i)$ é

$$C(i) = \frac{1}{n(n-1)} \sum_{s,t} \frac{\sigma_{st}(i)}{\sigma_{st}}. \tag{5.2}$$

Na equação (5.2) existem duas somas, uma nos vértices de origem s e nos de destino t. Para aplicar o algoritmo de *Metropolis*, teremos que separar estas duas somas. Para isso vamos definir as quantidades:

$$\delta_{st}(i) = \frac{\sigma_{st}(i)}{\sigma_{st}} \tag{5.3}$$

e

$$\delta_{s\bullet}(i) = \sum_t \delta_{st}(i). \tag{5.4}$$

Neste caso $C(i)$ torna-se

$$C(i) = \frac{1}{n(n-1)} \sum_s \delta_{s\bullet}(i). \tag{5.5}$$

5.2 Algoritmo *Metropolis* para o cálculo de $C(i)$

Os algoritmos conhecidos para calcular a intermediação possuem complexidade $\mathcal{O}(mn)$. Apesar desta complexidade ser polinomial, em muitos casos o cálculo exato de $C(i)$ ainda é inviável quando se trabalha com redes de tamanho médio com centenas de milhares de vértices.

Nesta seção, vamos utilizar o algoritmo *Metropolis* para estimar $C(i)$ e reduzir a complexidade para $\mathcal{O}(m)$. Para isso, na equação (5.2) vamos realizar apenas a soma em t e amostrar aleatoriamente vértices s e calcular a média. Este procedimento fornecerá uma $(\epsilon, \delta)-$aproximação para $C(i)$.

A amostragem dos vértices s é será feita com probabilidade

$$\mathrm{P}_i[s] = \frac{\delta_{s\bullet}(i)}{\sum_{s'} \delta_{s'\bullet}(i)}. \tag{5.6}$$

O denominador da equação (5.6) pode ser muito custoso de calcular se a rede for muito grande. Porém dado dois vértices s_1 e s_2, a razão $\mathrm{P}_i[s_1]/\mathrm{P}_i[s_2]$ é bem mais fácil de ser calculada e este fato será utilizado no algoritmo *Metropolis*.

5.2.1 Amostragem de *Metropolis*

Para amostrar T vértices, o amostrador MCMC segue os seguintes passos

- Escolha um vértice inicial s_0 de forma aleatória e uniforme.
- Para cada iteração j, $1 \leq j \leq T$, faça:
  - Seja s o estado atual da cadeia.
  - Escolha s' de forma aleatória e uniforme.
  - Com probabilidade $\min\left[1, \delta_{s'\bullet}(i)/\delta_{s\bullet}(i)\right]$ mude para o estado s'.

Seja M a lista dos vértices escolhidos nas transições da cadeia de Markov (incluindo vértices repetidos). Vamos agora encontrar uma aproximação $\tilde{C}(i)$ para $C(i)$ e em seguida mostrar dois teoremas que garantem a corretude da aproximação.

Seja $p(s)$ uma distribuição de probabilidades qualquer. Pela equação (5.5), podemos reescrever $C(i)$ como

$$C(i) = \frac{1}{n(n-1)} \sum_{s \in V} \frac{\delta_{s\bullet}(i)}{p(s)} p(s), \tag{5.7}$$

Considerando $p(s)$ como

$$p(s) = \mathrm{P}_i[s] = \frac{\delta_{s\bullet}(i)}{\sum_{s'} \delta_{s'\bullet}(i)}, \tag{5.8}$$

temos

$$C(i) \approx \frac{1}{n(n-1)} \frac{1}{T+1} \sum_{s \in M} \frac{\delta_{s\bullet}(i)}{p(s)}, \quad s \sim \mathrm{P}_i[s] \tag{5.9}$$

$$= \frac{1}{n(n-1)} \frac{1}{T+1} \sum_{s \in M} \sum_{s' \in V} \delta_{s'\bullet}(i), \quad s \sim \mathrm{P}_i[s]. \tag{5.10}$$

Seja $\overline{\delta(i)}$ a média dos $\delta_{s\bullet}(i)$, ou seja

$$\overline{\delta(i)} = \frac{1}{n} \sum_{s \in V} \delta_{s\bullet}(i), \tag{5.11}$$

então

$$C(i) \approx \frac{1}{n-1} \frac{1}{T+1} \sum_{s \in M} \overline{\delta(i)}, \quad s \sim \mathrm{P}_i[s] \tag{5.12}$$

Considerando

$$\sum_{s \in M} \overline{\delta(i)} \approx \sum_{s \in M} \delta_{s\bullet}(i), \tag{5.13}$$

temos

$$\tilde{C}(i) = \frac{1}{(T+1)(n-1)} \sum_{s \in M} \delta_{s\bullet}(i), \quad s \sim \mathrm{P}_i[s]. \tag{5.14}$$

Os dois teoremas a seguir garantem que $\tilde{C}(i)$ fornece uma estimativa arbitrariamente próxima de $C(i)$ com um numero polinomial (em $1/\epsilon$ e $\ln(\epsilon^{-1})$) de amostras.

Teorema 5.1. *Seja $\overline{\delta(i)}$ a média dos $\delta_{s\bullet}(i)$. Suponha que exista um $\mu(i)$ tal que para todo $s \in V$, a seguinte desigualdade seja válida*

$$\delta_{s\bullet}(i) \leq \mu(i) \overline{\delta(i)}. \tag{5.15}$$

Neste caso, para um $\epsilon > 0$, a probabilidade do desvio de $\tilde{C}(i)$ com relação a $C(i)$ ser maior que ϵ é

$$P\left[|\tilde{C}(i) - C(i)| > \epsilon\right] \leq 2 \exp\left\{-\frac{T}{2}\left(\frac{2\epsilon}{\mu(i)} - \frac{3}{T}\right)^2\right\}. \tag{5.16}$$

Prova. Ver [18]. □

Teorema 5.2. *A equação (5.14) fornece uma (ϵ, δ)-aproximação para $C(i)$ onde o número de amostras T é*

$$T \geq \frac{\mu(i)^2}{2\epsilon^2} \ln \frac{2}{\delta}. \tag{5.17}$$

Prova. Ver [18]. □

De acordo com [18], $\mu(i)$ pode ser considerado constante sob certas condições e a complexidade do pior caso do algoritmo é $\mathcal{O}(E(G))$.

CAPÍTULO 6

CONCLUSÃO

Neste trabalho foi apresentado a teoria geral de cadeias de Markov e método de Monte Carlo, com algumas aplicações incluindo a cálculo da centralidade de intermediação.

Verificou-se que o método de Monte Carlo e o método MCMC, por serem algoritmos de amostragem de caráter muito geral, possuem inúmeras aplicações como visto nos exemplos citados.

Para uma cadeia de Markov, foi demostrado que o tempo de mistura é uma variável de extrema importância, pois ela é uma das principais variáveis que ditam se o algoritmo é eficiente ou não. Também foram mostradas as principais técnicas utilizadas para limitar o tempo de mistura.

Por último, dada a importância das medidas de centralidade em redes complexas, o cálculo da centralidade de intermediação foi escolhida para a aplicação do algoritmo *Metropolis*. Verificou-se que para redes com milhares de vértices, é possível estimar a centralidade de intermediação realizando uma pequena amostragem no conjunto de vértices, obtendo um erro que decresce exponencialmente com o número de amostras.

APÊNDICE A

DECOMPOSIÇÃO ESPECTRAL

A.1 Autovalores e autovetores

Considere a matriz $\mathbf{A} \in M_n(\mathbb{R})$ e a seguinte equação de autovalor

$$\mathbf{A}\mathbf{v}_i = \lambda_i \mathbf{v}_i. \tag{A.1}$$

O vetor $\mathbf{v}_i$ é chamado de autovetor de $\mathbf{A}$ e λ_i é o autovalor correspondente. Para encontrar os autovalores λ_i, devemos encontrar as raízes do polinômio

$$p(x) = \det(\mathbf{A} - \mathbb{1}x) = 0, \tag{A.2}$$

que é chamado de polinômio característico. Podemos fatorar o polinômio característico da seguinte forma

$$p(x) = (x - \lambda_1)\dots(x - \lambda_n) = (x - \lambda_1)^{m_1}\dots(x - \lambda_k)^{m_k}, \quad k \leq n. \tag{A.3}$$

O número de vezes m_i que o fator $x - \lambda_i$ aparece em $p(x)$ é chamado de multiplicidade algébrica de λ_i.

A.2 Teorema espectral

Teorema A.1. *Seja* $\mathbf{A}$ *uma matriz simétrica com coeficientes reais. Então*

(i) Os autovalores de $\mathbf{A}$ *são reais.*

(ii) Os autovetores de $\mathbf{A}$ *correspondentes a diferentes autovetores são ortogonais.*

Prova: (i): Considere $\mathbf{v}_i$ está normalizado, ou seja, $\langle \mathbf{v}_i, \mathbf{v}_i \rangle = 1$, onde $\langle \cdot, \cdot \rangle$ denota o produto escalar.

$$\lambda_i = \lambda_i \langle \mathbf{v}_i, \mathbf{v}_i \rangle = \langle \lambda_i \mathbf{v}_i, \mathbf{v}_i \rangle = \langle \mathbf{A}\mathbf{v}_i, \mathbf{v}_i \rangle = \langle \mathbf{v}_i, \mathbf{A}^\top \mathbf{v}_i \rangle \tag{A.4}$$

$$= \langle \mathbf{v}_i, \mathbf{A}\mathbf{v}_i \rangle = \langle \mathbf{v}_i, \lambda_i \mathbf{v}_i \rangle = \langle \lambda_i \mathbf{v}_i, \mathbf{v}_i \rangle^* = \lambda_i^* \langle \mathbf{v}_i, \mathbf{v}_i \rangle \tag{A.5}$$

$$\lambda_i = \lambda_i^*. \tag{A.6}$$

(ii): Considere o produto:

$$\langle \mathbf{A}\mathbf{v}_i, \mathbf{v}_j \rangle = \mathbf{v}_i^\top \mathbf{A}^\top \mathbf{v}_j = \mathbf{v}_i^\top (\mathbf{A}\mathbf{v}_j) = \lambda_j \mathbf{v}_i^\top \mathbf{v}_j = \lambda_j \langle \mathbf{v}_i, \mathbf{v}_j \rangle \tag{A.7}$$

por outro lado

$$\langle \mathbf{A}\mathbf{v}_i, \mathbf{v}_j \rangle = \mathbf{v}_i^\top \mathbf{A}^\top \mathbf{v}_j = (\mathbf{A}\mathbf{v}_i)^\top \mathbf{v}_j = \lambda_i \mathbf{v}_i^\top \mathbf{v}_j = \lambda_i \langle \mathbf{v}_i, \mathbf{v}_j \rangle \tag{A.8}$$

subtraindo estas duas equações, temos

$$(\lambda_i - \lambda_j)\langle \mathbf{v}_i, \mathbf{v}_j \rangle = 0. \tag{A.9}$$

Portanto se $\lambda_i \neq \lambda_j \rightarrow \langle \mathbf{v}_i, \mathbf{v}_j \rangle = 0$. □

Definição A.1. *Uma matriz* $\mathbf{C}$ *é ortogonal se* $\mathbf{C}\mathbf{C}^\top = \mathbf{C}^\top \mathbf{C} = \mathbb{1}$.

Perceba que $\mathbf{C}^{-1} = \mathbf{C}^\top$. Além disso, uma matriz $n \times n$ onde cada coluna é uma base ortonormal em $\mathbb{R}^n$ é uma matriz ortogonal.

Teorema A.2. *(Teorema espectral) Seja* $\mathbf{A} \in M_n(\mathbb{R})$ *uma matriz simétrica. Existe uma matriz ortogonal* $\mathbf{C}$ *cujas colunas são os autovetores* $\mathbf{v}_i$ *de* $\mathbf{A}$ *e que formam uma base de* $\mathbb{R}^n$ *de modo que a matriz* $\mathbf{C}^{-1}\mathbf{A}\mathbf{C}$ *é uma matriz diagonal composta dos autovalores* λ_i*, ou seja* $\mathbf{C}^{-1}\mathbf{A}\mathbf{C} = diag(\lambda_1, \ldots, \lambda_n)$.

Prova: Ver [19]. □

Corolário A.1. *A dimensão do subespaço associado a um autovalor é igual à multiplicidade deste autovalor.*

Prova: Ver [19]. □

Corolário A.2. *A matriz* $\mathbf{A}$ *pode ser escrita como* $\mathbf{A} = \sum_{i=1}^n \lambda_i \mathbf{v}_i \mathbf{v}_i^\top$.

Prova: Vamos primeiro mostrar que $\sum_{i=1}^n \mathbf{v}_i \mathbf{v}_i^\top = \mathbb{1}$.

Considere $\mathbf{V}$ uma matriz onde as colunas são autovetores linearmente independentes da matriz $\mathbf{A}$. Assim temos

$$\mathbb{1} = \mathbf{V}\mathbf{V}^\top. \tag{A.10}$$

O elemento $\mathbb{1}_{i,j}$ é

$$\mathbb{1}_{i,j} = \sum_{k=1}^n V_{i,k} V_{k,j}^\top = \sum_{k=1}^n (\mathbf{v}_k)_i (\mathbf{v}_k^\top)_j, \tag{A.11}$$

onde $(\mathbf{v}_k)_i$ indica a $i-$ésima componente do autovetor $\mathbf{v}_k$. Portanto a matriz $\mathbb{1}$ fica

$$\mathbb{1} = \sum_{k=1}^n \begin{bmatrix} (\mathbf{v}_k)_1 \\ (\mathbf{v}_k)_2 \\ \vdots \\ (\mathbf{v}_k)_n \end{bmatrix} \begin{bmatrix} (\mathbf{v}_k^\top)_1 & (\mathbf{v}_k^\top)_2 & \ldots & (\mathbf{v}_k^\top)_n \end{bmatrix} = \sum_{i=1}^n \mathbf{v}_i \mathbf{v}_i^\top. \tag{A.12}$$

Vamos agora aplicar este operador à esquerda e direita de $\mathbf{A}$

$$\mathbf{A} = \mathbb{1}\mathbf{A}\mathbb{1} = \left(\sum_{i=1}^{n} \mathbf{v}_i \mathbf{v}_i^\top\right) \mathbf{A} \left(\sum_{j=1}^{n} \mathbf{v}_j \mathbf{v}_j^\top\right) = \sum_{i,j} \mathbf{v}_i \mathbf{v}_i^\top \mathbf{A} \mathbf{v}_j \mathbf{v}_j^\top \tag{A.13}$$

$$\mathbf{A} = \sum_{i,j} \mathbf{v}_i \mathbf{v}_i^\top \lambda_j \mathbf{v}_j \mathbf{v}_j^\top = \sum_{i,j} \lambda_j \mathbf{v}_i (\mathbf{v}_i^\top \mathbf{v}_j) \mathbf{v}_j^\top = \sum_{i,j} \lambda_j \delta_{i,j} \mathbf{v}_i \mathbf{v}_j^\top \tag{A.14}$$

$$\mathbf{A} = \sum_{i=1}^{n} \lambda_i \mathbf{v}_i \mathbf{v}_i^\top . \tag{A.15}$$

□

APÊNDICE B

SIMETRIZAÇÃO DA MATRIZ ESTOCÁSTICA E CONVERGÊNCIA PARA O EQUILÍBRIO

B.1 Simetrização de P

Proposição B.1. *Seja* $\mathbf{P}$ *a matriz estocástica de uma cadeia de Markov reversível e ergódica com distribuição estacionária* $\bar{\pi} = (\pi_1, \dots \pi_n)$. *Seja* $\mathbf{M}$ *uma matriz definida por* $\mathbf{M} = \mathbf{\Lambda P \Lambda}^{-1}$, *onde* $\mathbf{\Lambda} = diag(\sqrt{\pi_1}, \sqrt{\pi_2}, \dots, \sqrt{\pi_n})$, *ou seja, a* $i-$*ésima diagonal de* $\mathbf{\Lambda}$ *é* $\sqrt{\pi_i}$. *Então*

(i) $\mathbf{M}$ *é simétrica.*

(ii) $\mathbf{M}$ *e* $\mathbf{P}$ *possuem os mesmos autovalores.*

Prova: (i) Considere o elemento $M_{i,j}$ de $\mathbf{M}$

$$M_{i,j} = \sum_{k,l} \Lambda_{i,k} P_{k,l} \Lambda^{-1}_{l,j} \tag{B.1}$$

Como a matriz $\mathbf{\Lambda}$ é diagonal, podemos escrever $\Lambda_{i,k} = \sqrt{\pi_i}\delta_{i,k}$ e $\Lambda^{-1}_{l,j} = \delta_{l,j}/\sqrt{\pi_j}$, onde $\delta_{i,j}$ é a função delta de Kronecker definida como: $\delta_{i,j} = 1$ se $i = j$ e $\delta_{i,j} = 0$ caso contrário . Portanto

$$M_{i,j} = \sum_{k,l} \sqrt{\pi_i}\delta_{i,k} P_{k,l} \frac{\delta_{l,j}}{\sqrt{\pi_j}} = \sqrt{\frac{\pi_i}{\pi_j}} P_{i,j}. \tag{B.2}$$

Como a cadeia é reversível temos que $\pi_i P_{i,j} = \pi_j P_{j,i}$, ou $P_{i,j} = P_{j,i}\pi_j/\pi_i$, então

$$M_{i,j} = \sqrt{\frac{\pi_i}{\pi_j}\frac{\pi_j}{\pi_i}} P_{j,i} = \sqrt{\frac{\pi_j}{\pi_i}} P_{j,i} = M_{j,i}. \tag{B.3}$$

(ii) Considere que a matriz $\mathbf{M}$ possui autovalores λ_i e respectivos autovetores $\mathbf{v}_i$, ou seja, $\mathbf{M}\mathbf{v}_i = \lambda_i \mathbf{v}_i$, $1 \leq i \leq n$.

Vamos aplicar a matriz $\mathbf{P}$ sobre o vetor $\Psi_i^R = \mathbf{\Lambda}^{-1}\mathbf{v}_i$:

$$\mathbf{P}(\mathbf{\Lambda}^{-1}\mathbf{v}_i) = \mathbf{\Lambda}^{-1}\mathbf{\Lambda}\mathbf{P}\mathbf{\Lambda}^{-1}\mathbf{v}_i = \mathbf{\Lambda}^{-1}(\mathbf{\Lambda}\mathbf{P}\mathbf{\Lambda}^{-1})\mathbf{v}_i = \mathbf{\Lambda}^{-1}\mathbf{M}\mathbf{v}_i = \lambda_i(\mathbf{\Lambda}^{-1}\mathbf{v}_i). \tag{B.4}$$

Portanto λ_i é autovalor de $\mathbf{P}$ com respectivo autovetor a direita Ψ_i^R. Além disso, $\mathbf{P}$ possui autovetor a esquerda correspondente a λ_i dado por $\Psi_i^L = \mathbf{v}_i^\top\mathbf{\Lambda}$, pois

$$(\mathbf{v}_i^\top\mathbf{\Lambda})\mathbf{P} = \mathbf{v}_i^\top\mathbf{\Lambda}\mathbf{P}\mathbf{\Lambda}^{-1}\mathbf{\Lambda} = \mathbf{v}_i^\top(\mathbf{\Lambda}\mathbf{P}\mathbf{\Lambda}^{-1})\mathbf{\Lambda} = \mathbf{v}_i^\top\mathbf{M}\mathbf{\Lambda} = (\mathbf{M}\mathbf{v}_i)^\top\mathbf{\Lambda} = \lambda_i(\mathbf{v}_i^\top\mathbf{\Lambda}) \tag{B.5}$$

□

Proposição B.2. *Seja* $\mathbf{1}$ *um vetor coluna de 'uns', então* $\Psi_1^R = \mathbf{1}$ *e* $\Psi_1^L = \bar{\pi}$.

Prova: Considere a aplicação de $\mathbf{P}$ sobre $\mathbf{1}$

$$\mathbf{P1} = \begin{pmatrix} P_{1,1} & P_{1,2} & \cdots & P_{1,j} \\ P_{2,1} & P_{2,2} & \cdots & P_{2,j} \\ \vdots & \vdots & \ddots & \vdots \\ P_{i,1} & P_{i,1} & \cdots & P_{n,n} \end{pmatrix} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} = \begin{bmatrix} \sum_j P_{1,j} \\ \sum_j P_{2,j} \\ \vdots \\ \sum_j P_{n,j} \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}. \tag{B.6}$$

Portanto $\mathbf{1}$ é autovetor a direita com autovalor 1. Além disso sabemos que $\bar{\pi}$ é autovetor a esquerda com autovalor 1, portanto $\bar{\pi} = \Psi_1^L$. □

B.2 Convergência para o equilíbrio

Proposição B.3. *Seja* $\mathbf{P}$ *a matriz estocástica de uma cadeia de Markov reversível e ergódica e sejam* $\mathbf{p}(0)$ *a distribuição inicial,* Ψ_i^R *e* Ψ_i^L *os autovetores a direita e a esquerda respectivamente da matriz estocástica* $\mathbf{P}$. *Então* $\mathbf{P}^t = \sum_{i=1}^n \lambda_i^t\Psi_i^R\Psi_i^L$ *e* $\mathbf{p}(t) = \bar{\pi} + \sum_{i=2}^n \lambda_i^t\langle\mathbf{p}(0), \Psi_i^R\rangle\Psi_i^L$.

Prova: Como $\mathbf{M} = \mathbf{\Lambda}\mathbf{P}\mathbf{\Lambda}^{-1}$ é simétrica, podemos escrevê-la como (ver apêndice A).

$$\mathbf{M} = \sum_{i=1}^n \lambda_i\mathbf{v}_i\mathbf{v}_i^\top. \tag{B.7}$$

Como $\mathbf{P} = \mathbf{\Lambda}^{-1}\mathbf{M}\mathbf{\Lambda}$, $\mathbf{P}^t$ será:

$$\mathbf{P}^t = (\mathbf{\Lambda}^{-1}\mathbf{M}\mathbf{\Lambda})\dots(\mathbf{\Lambda}^{-1}\mathbf{M}\mathbf{\Lambda}) = \mathbf{\Lambda}^{-1}\mathbf{M}^t\mathbf{\Lambda}. \tag{B.8}$$

$$= \mathbf{\Lambda}^{-1}\left(\sum_{i=1}^n \lambda_i\mathbf{v}_i\mathbf{v}_i^\top\right)^t\mathbf{\Lambda} = \mathbf{\Lambda}^{-1}\left(\sum_{i_1=1}^n \lambda_{i_1}\mathbf{v}_{i_1}\mathbf{v}_{i_1}^\top\right)\dots\left(\sum_{i_t=1}^n \lambda_{i_t}\mathbf{v}_{i_t}\mathbf{v}_{i_t}^\top\right)\mathbf{\Lambda} = \mathbf{\Lambda}^{-1}\sum_{i=1}^n \lambda_i^t\mathbf{v}_i\mathbf{v}_i^\top\mathbf{\Lambda} \tag{B.9}$$

$$= \sum_{i=1}^n \lambda_i^t(\mathbf{\Lambda}^{-1}\mathbf{v}_i)(\mathbf{v}_i^\top\mathbf{\Lambda}) = \sum_{i=1}^n \lambda_i^t\Psi_i^R\Psi_i^L \tag{B.10}$$

$$\mathbf{P}^t = \mathbf{1}\bar{\pi} + \sum_{i=2}^n \lambda_i^t\Psi_i^R\Psi_i^L. \tag{B.11}$$

A matriz $\mathbf{1}\bar{\pi}$ possui todas as linhas iguais. Cada linha é igual à distribuição estacionária, ou seja, $(\mathbf{1}\bar{\pi})_{i,j} = \pi_j = \lim_{t\to\infty} P_{i,j}^t$.

Se a distribuição de probabilidades em $t = 0$ é $\mathbf{p}(0)$, então no tempo t temos

$$\mathbf{p}(t) = \mathbf{p}(0)P^t = \langle \mathbf{p}(0), \mathbf{1} \rangle \bar{\pi} + \sum_{i=2}^{n} \lambda_i^t \langle \mathbf{p}(0), \Psi_i^R \rangle \Psi_i^L. \tag{B.12}$$

O termo $\langle \mathbf{p}(0), \mathbf{1} \rangle$ é a soma dos elementos de $\mathbf{p}(0)$, que vale 1 e o termo $\langle \mathbf{p}(0), \Psi_i^R \rangle$, que chamaremos de $\alpha_i(0)$, é a projeção do estado inicial nos vetores da base Ψ_R. Assim $\mathbf{p}(t)$ é

$$\mathbf{p}(t) = \bar{\pi} + \sum_{i=2}^{n} \lambda_i^t \alpha_i(0) \Psi_i^L. \tag{B.13}$$

Como $1 = \lambda_1 > \lambda_2 \geq \lambda_3 \geq \cdots \geq \lambda_n \geq -1$, a velocidade de convergência da cadeia depende do maior autovalor $\lambda_{max} = \max\{|\lambda_i|; 1 < i \leq n\}$

$$\mathbf{p}(t) = \bar{\pi} + \mathcal{O}(\lambda_{max}^t) \sum_{i=2}^{n} \alpha_i(0) \Psi_i^L. \tag{B.14}$$

□

APÊNDICE C

LAPLACIANO E CONECTIVIDADE DE UM GRAFO

C.1 Laplaciano

Seja $G = (V, E)$ um grafo não direcionado, $\mathbf{A}$ a sua matriz de adjacência e $\mathbf{D}$ uma matriz diagonal onde a $i-$ésima diagonal é o grau do vértice i, ou seja $\mathbf{D} = \text{diag}(d_1, \ldots, d_n)$. O laplaciano de G é definido como

$$\mathbf{L} = \mathbf{D} - \mathbf{A}. \tag{C.1}$$

A seguir vamos calcular o valor de $\langle \mathbf{x}, \mathbf{Lx} \rangle = \mathbf{x}^\top \mathbf{Lx}$ pois esta quantidade será muito utilizada neste trabalho.

Proposição C.1. *Seja* $\mathbf{x} = [x_1, \ldots, x_n]^\top$ *um vetor coluna qualquer e* $\mathbf{L}$ *o laplaciano de um grafo não direcionado* G*. Então*

$$\mathbf{x}^\top \mathbf{Lx} = \sum_{\{i,j\} \in E} (x_i - x_j)^2. \tag{C.2}$$

Prova:

$$\mathbf{x}^\top \mathbf{Lx} = \mathbf{x}^\top \mathbf{Dx} - \mathbf{x}^\top \mathbf{Ax} = \sum_i x_i d_i x_i - \sum_{i,j} x_i A_{i,j} x_j. \tag{C.3}$$

mas $d_i = \sum_j A_{i,j}$, então

$$\mathbf{x}^\top \mathbf{Lx} = \sum_{i,j} A_{i,j} x_i^2 - \sum_{i,j} x_i A_{i,j} x_j = \frac{1}{2} \left(\sum_{i,j} A_{i,j} x_i^2 + \sum_{i,j} A_{i,j} x_i^2 \right) - \sum_{i,j} x_i A_{i,j} x_j \tag{C.4}$$

$$= \frac{1}{2} \left(\sum_{i,j} A_{i,j} x_i^2 + \sum_{j,i} A_{j,i} x_j^2 \right) - \sum_{i,j} x_i A_{i,j} x_j = \frac{1}{2} \sum_{i,j} A_{i,j} (x_i^2 + x_j^2) - \sum_{i,j} x_i A_{i,j} x_j \tag{C.5}$$

$$= \sum_{\{i,j\} \in E} (x_i^2 + x_j^2 - 2 x_i x_j) \tag{C.6}$$

$$\mathbf{x}^\top \mathbf{L}\mathbf{x} = \sum_{\{i,j\}\in E} (x_i - x_j)^2. \tag{C.7}$$

□

Proposição C.2. *Seja* $\mathbf{L}$ *o laplaciano de um grafo não direcionado* G*. Sejam* $\mathbf{w}_1, \ldots, \mathbf{w}_n$ *seus autovetores com respectivos autovalores* $\mu_1 \leq \cdots \leq \mu_n$*. Então*

(i) Os autovalores da matriz $\mathbf{L}$ *não são negativos.*

(ii) A matriz $\mathbf{L}$ *possui um autovalor nulo com respectivo autovetor constante (todas as componentes do vetor são iguais).*

Prova: (i): Pela equação (C.7), vemos que $\mathbf{x}^\top \mathbf{L}\mathbf{x} \geq 0$. Para um autovetor $\mathbf{w}_i$ de $\mathbf{L}$ temos

$$\mathbf{w}_i^\top \mathbf{L}\mathbf{w}_i = \mu_i \mathbf{w}_i^\top \mathbf{w}_i \geq 0 \Rightarrow \mu_i \geq 0. \tag{C.8}$$

Neste caso dizemos que a matriz $\mathbf{L}$ é positiva semi-definida.

(ii): Considere um vetor constante $\mathbf{x} = [1, \ldots, 1]^\top$.

$$\mathbf{L}\mathbf{x} = \mathbf{D}\mathbf{x} - \mathbf{A}\mathbf{x} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix} - \begin{bmatrix} \sum_j A_{1,j} \\ \sum_j A_{2,j} \\ \vdots \\ \sum_j A_{n,j} \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix} - \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix} = \mathbf{0} = 0\mathbf{x}. \tag{C.9}$$

Portanto $\mu_1 = 0$ e $\mathbf{w}_1 = [1, \ldots, 1]^\top$. □

Teorema C.1. *Sejam* $G = (V, E)$ *um grafo não direcionado,* $\mathbf{L}$ *o laplaciano de* G *e* $0 = \mu_1 \leq \mu_2 \leq \cdots \leq \mu_n$ *os autovalores de* $\mathbf{L}$*. Então* $\mu_2 > 0$ *se e somente se* G *é conexo.*

Prova:

- $\mu_2 > 0 \Rightarrow G$ é conexo.

Pela contra positiva, vamos provar que G desconexo $\Rightarrow \mu_2 = 0$. Considere que G é desconexo. Neste caso podemos dividir G em duas componentes G_1 e G_2 e fazer uma renomeação dos vértices $V = \{v'_1, \ldots v'_k, v'_{k+1}, \ldots, v'_n\}$, onde os índices de 1 a k são dos vértices de G_1 e os índices de $k+1$ a n são dos vértices de G_2. Assim o laplaciano de G será uma matriz em blocos da forma

$$\mathbf{L} = \begin{bmatrix} \mathbf{L}_{G_1} & \mathbf{0} \\ \mathbf{0} & \mathbf{L}_{G_2} \end{bmatrix}. \tag{C.10}$$

Se encontrarmos dois autovetores de $\mathbf{L}$ que são linearmente independentes e que possuem o mesmo autovalor 0, então teremos provado que μ_2 também é nulo.

Considere os vetores coluna compostos de duas partes

$$\mathbf{z}_1 = \begin{bmatrix} \mathbf{1} \\ \mathbf{0} \end{bmatrix} \quad \text{e} \quad \mathbf{z}_2 = \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \end{bmatrix}, \tag{C.11}$$

onde o índice da primeira parte vai de 1 a k e da segunda de $k+1$ a n.

Fazendo o produto escalar $\langle \mathbf{z}_1, \mathbf{z}_2 \rangle$, verificamos que $\mathbf{z}_1$ e $\mathbf{z}_2$ são linearmente independentes. Além disso ambos possuem autovalor nulo, pois $\mathbf{L}\mathbf{z}_1 = \mathbf{0} = \mathbf{L}\mathbf{z}_2$ e portanto $\mu_2 = 0$.

- G é conexo $\Rightarrow \mu_2 > 0$.

Considere G conexo e considere $\mathbf{x}$ um autovetor de $\mathbf{L}$ com autovalor zero. Assim temos

$$\mathbf{Lx} = \mathbf{0} \tag{C.12}$$

$$\mathbf{x}^\top \mathbf{Lx} = 0. \tag{C.13}$$

Pela equação (C.7) temos

$$\sum_{\{i,j\} \in E} (x_i - x_j)^2 = 0. \tag{C.14}$$

Portanto para cada par i, j, onde $\{i, j\} \in E$ devemos ter $x_i = x_j$. Como G é conexo, temos indutivamente que $x_1 = x_2 = \cdots = x_n$, ou seja, a única possibilidade para $\mathbf{x}$ é ser um vetor constante. Assim o autoespaço correspondente ao autovalor zero tem dimensão 1 e $\mu_2 > 0$. □

APÊNDICE D

DESIGUALDADE DE CHEEGER

D.1 Laplaciano normalizado

Definição D.1 (Laplaciano normalizado)**.** *Seja* $\mathbf{A}$ *a matriz de adjacência de um grafo* G*. Seja* $\mathbf{D}^{1/2} = diag(\sqrt{d_1}, \sqrt{d_2}, ..$ *onde* d_i *é o grau do vértice* i*. O laplaciano normalizado de* G *é definido por* $\mathcal{L} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2} = \mathbb{1} - \tilde{\mathbf{A}}$*, onde* $\tilde{\mathbf{A}} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ *é a matriz de adjacência normalizada.*

Proposição D.1. *Sejam* $\mu_1 \leq \mu_2 \leq \cdots \leq \mu_n$ *os autovalores de* $\mathcal{L}$ *com respectivos autovetores ortogonais* $\mathbf{w}_1, \mathbf{w}_2, \ldots \mathbf{w}_n$*. Então*

(i) Os autovalores de $\mathcal{L}$ *são maiores ou iguais a zero.*

(ii) O primeiro autovalor μ_1 *é nulo com respectivo autovetor* $\mathbf{D}^{1/2}\mathbf{1}$*.*

(iii) O segundo autovalor μ_2 *é nulo se e somente se* G *é desconexo.*

Prova: (i) Considere dois vetores quaisquer $\mathbf{x}$ e $\mathbf{y}$ tal que $\mathbf{x} = \mathbf{D}^{1/2}\mathbf{y}$. Vamos calcular $\mathbf{x}^\top \mathcal{L}\mathbf{x}$

$$\mathbf{x}^\top \mathcal{L}\mathbf{x} = \mathbf{x}^\top \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}\mathbf{x} = \mathbf{y}^\top \mathbf{L}\mathbf{y} = \sum_{\{i,j\}\in E} (y_i - y_j)^2 \geq 0. \tag{D.1}$$

Onde a última igualdade foi obtida de (C.7). Considere que $\mathbf{x} = \mathbf{w}_i$, neste caso

$$\mathbf{w}_i^\top \mathcal{L}\mathbf{w}_i = \mu_i \mathbf{w}_i^\top \mathbf{w}_i \geq 0 \quad \Rightarrow \quad \mu_i \geq 0. \tag{D.2}$$

(ii) Vamos aplicar $\mathcal{L}$ sobre o vetor $\mathbf{z} = \mathbf{D}^{1/2}\mathbf{1}$, onde $\mathbf{1}$ é um vetor de 'uns'.

$$\mathcal{L}(\mathbf{D}^{1/2}\mathbf{1}) = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}\mathbf{D}^{1/2}\mathbf{1} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{1} = \mathbf{0}. \tag{D.3}$$

Portanto $\mathbf{z} = \mathbf{w}_1$ e $\mu_1 = 0$.

(iii) Basta seguir o mesmo procedimento do teorema C.1 dos apêndices. □

D.2 Quociente de Rayleigh

Definição D.2. *O quociente de Rayleigh para uma matriz hermitiana* $\mathbf{M}$ *e vetor* $\mathbf{x}$ *é definida por*

$$\mathcal{R}(\mathbf{M}, \mathbf{x}) = \frac{\mathbf{x}^\top \mathbf{M}\mathbf{x}}{\mathbf{x}^\top \mathbf{x}}. \tag{D.4}$$

Proposição D.2. *Sejam* $\mathbf{z}_1, \ldots, \mathbf{z}_n$ *os autovetores de* $\mathbf{M}$ *com respectivos autovalores* $\alpha_1 \leq \cdots \leq \alpha_n$*, então*

(i) $\alpha_1 = \min_{\mathbf{x}} \mathcal{R}(\mathbf{M}, \mathbf{x}) = \min_{\mathbf{x}} \frac{\mathbf{x}^\top \mathbf{M}\mathbf{x}}{\mathbf{x}^\top \mathbf{x}}$.

(ii) $\alpha_2 = \min_{\mathbf{x}; \mathbf{x} \perp \mathbf{z}_1} \mathcal{R}(\mathbf{M}, \mathbf{x}) = \min_{\mathbf{x}; \mathbf{x} \perp \mathbf{z}_1} \frac{\mathbf{x}^\top \mathbf{M}\mathbf{x}}{\mathbf{x}^\top \mathbf{x}}$.

(iii) $\alpha_n = \max_{\mathbf{x}} \mathcal{R}(\mathbf{M}, \mathbf{x}) = \max_{\mathbf{x}} \frac{\mathbf{x}^\top \mathbf{M}\mathbf{x}}{\mathbf{x}^\top \mathbf{x}}$.

Prova: (i) Considere a decomposição de $\mathbf{x}$ nos autovetores de $\mathbf{M}$: $\mathbf{x} = \sum_i a_i \mathbf{z}_i$. $\mathcal{R}(\mathbf{M}, \mathbf{x})$ é

$$\mathcal{R}(\mathbf{M}, \mathbf{x}) = \frac{\sum_i a_i \mathbf{z}_i^\top \mathbf{M} \sum_j a_j \mathbf{z}_j}{\sum_{i,j} a_i \mathbf{z}_i^\top a_j \mathbf{z}_j} = \frac{\sum_i a_i^2 \alpha_i}{\sum_i a_i^2} \geq \alpha_1 \frac{\sum_i a_i^2}{\sum_i a_i^2} = \alpha_1, \tag{D.5}$$

onde o mínimo de $\mathcal{R}(\mathbf{M}, \mathbf{x})$ é obtido escolhendo $\mathbf{x} = \mathbf{z}_1$.

(ii) Considere que $\mathbf{x} \perp \mathbf{z}_1$ e $\mathbf{x} = \sum_{i \neq 1} a_i \mathbf{z}_i$. $\mathcal{R}(\mathbf{M}, \mathbf{x})$ é

$$\mathcal{R}(\mathbf{M}, \mathbf{x}) = \frac{\sum_{i \neq 1} a_i \mathbf{z}_i^\top \mathbf{M} \sum_{j \neq 1} a_j \mathbf{z}_j}{\sum_{i \neq 1, j \neq 1} a_i \mathbf{z}_i^\top a_j \mathbf{z}_j} = \frac{\sum_{i \neq 1} a_i^2 \alpha_i}{\sum_{i \neq 1} a_i^2} \geq \alpha_2 \frac{\sum_i a_i^2}{\sum_i a_i^2} = \alpha_2, \tag{D.6}$$

onde o mínimo de $\mathcal{R}(\mathbf{M}, \mathbf{x})$ é obtido fazendo $\mathbf{x} = \mathbf{z}_2$.

(iii) Semelhante à prova de 1. □

D.3 Prova da desigualdade de Cheeger (lado esquerdo)

Teorema D.1. *(Desigualdade de Cheeger) Seja* μ_2 *o segundo menor autovalor de* $\mathcal{L}$*, então*

$$\frac{\mu_2}{2} \leq \Phi \leq \sqrt{2\mu_2} \tag{D.7}$$

ou de forma equivalente

$$\frac{\Phi^2}{2} \leq \mu_2 \leq 2\Phi. \tag{D.8}$$

Prova de $\mu_2/2 \leq \Phi$*:* Pelas propriedades do quociente de Rayleigh, temos

$$\mu_2 = \min_{\mathbf{x}; \mathbf{x} \perp \mathbf{w}_1} \frac{\mathbf{x}^\top \mathcal{L}\mathbf{x}}{\mathbf{x}^\top \mathbf{x}} = \min_{\mathbf{x}; \mathbf{x} \perp \mathbf{w}_1} \frac{\mathbf{x}^\top \mathbf{D}^{-1/2} \mathbf{L} \mathbf{D}^{-1/2} \mathbf{x}}{\mathbf{x}^\top \mathbf{x}} = \min_{(\mathbf{y}; \mathbf{D}^{1/2}\mathbf{y} \perp \mathbf{D}^{1/2}\mathbf{1})} \frac{\mathbf{y}^\top \mathbf{L}\mathbf{y}}{\mathbf{y}^\top \mathbf{D}\mathbf{y}}, \tag{D.9}$$

onde $\mathbf{y} = \mathbf{D}^{-1/2}\mathbf{x}$ e $\mathbf{w}_1 = \mathbf{D}^{1/2}\mathbf{1}$, como visto no início da seção.

A restrição $\mathbf{x} \perp \mathbf{w}_1$ é equivalente a

$$\mathbf{x} \perp \mathbf{w}_1 \iff \mathbf{D}^{1/2}\mathbf{y} \perp \mathbf{D}^{1/2}\mathbf{1} \iff \langle \mathbf{D}^{1/2}\mathbf{y}, \mathbf{D}^{1/2}\mathbf{1} \rangle = 0 \iff \sum_i y_i d_i = 0. \tag{D.10}$$

Substituindo (C.7) em (D.9) e usando (D.10) temos

$$\mu_2 = \min_{\mathbf{y};\sum_i y_i d_i = 0} \frac{\sum_{\{i,j\}\in E}(y_i - y_j)^2}{\sum_i y_i^2 d_i}. \tag{D.11}$$

O vetor $\mathbf{y}$ pode ser encarado como uma função que retorna um valor y_i dado o vértice i. Desejamos provar a equação acima é menor que 2Φ. Para isso, é suficiente encontrar um $\mathbf{y}$ qualquer tal que a equação acima sem a minimização seja menor que 2Φ.

Para encontrar um $\mathbf{y}$, vamos definir o conjunto $S^\star$ tal que $\Phi(S^\star)$ é mínima, ou seja $\Phi(S^\star) = \Phi$. Vamos definir $\mathbf{y}$ como

$$y_i = \begin{cases} \frac{1}{\mathrm{Vol}(S^\star)}, & \text{se } i \in S^\star \\ -\frac{1}{\mathrm{Vol}(V - S^\star)}, & \text{se } i \notin S^\star \end{cases}. \tag{D.12}$$

É fácil ver que $\mathbf{y}$ obedece a restrição $\sum_i y_i d_i = 0$, pois

$$\sum_i y_i d_i = \sum_{i\in S^\star} y_i d_i + \sum_{i\notin S^\star} y_i d_i = \frac{\sum_{i\in S^\star} d_i}{\mathrm{Vol}(S^\star)} - \frac{\sum_{i\notin S^\star} d_i}{\mathrm{Vol}(V - S^\star)} = 0. \tag{D.13}$$

Na equação ($D.11$), os únicos termos do numerador que contribuem para a soma são as arestas que possuem uma ponta em $S^\star$ e a outra em $V - S^\star$. Assim temos

$$\mu_2 \leq \frac{\sum_{\{i,j\}\in E}(y_i - y_j)^2}{\sum_i y_i^2 d_i} = \frac{|E(S^\star, V - S^\star)| \left(\frac{1}{\mathrm{Vol}(S^\star)} + \frac{1}{\mathrm{Vol}(V - S^\star)}\right)^2}{\sum_{i\in S^\star} d_i \left(\frac{1}{\mathrm{Vol}(S^\star)}\right)^2 + \sum_{i\notin S^\star} d_i \left(\frac{1}{\mathrm{Vol}(V - S^\star)}\right)^2} \tag{D.14}$$

$$= |E(S^\star, V - S^\star)| \left(\frac{1}{\mathrm{Vol}(S^\star)} + \frac{1}{\mathrm{Vol}(V - S^\star)}\right)^2 / \left(\frac{1}{\mathrm{Vol}(S^\star)} + \frac{1}{\mathrm{Vol}(V - S^\star)}\right) \tag{D.15}$$

$$= |E(S^\star, V - S^\star)| \left(\frac{1}{\mathrm{Vol}(S^\star)} + \frac{1}{\mathrm{Vol}(V - S^\star)}\right) \tag{D.16}$$

$$\leq 2|E(S^\star, V - S^\star)| \max\left(\frac{1}{\mathrm{Vol}(S^\star)}, \frac{1}{\mathrm{Vol}(V - S^\star)}\right) \tag{D.17}$$

$$= \frac{2|E(S^\star, V - S^\star)|}{\min\left(\mathrm{Vol}(S^\star), \mathrm{Vol}(V - S^\star)\right)} \tag{D.18}$$

$$\mu_2 \leq 2\Phi. \tag{D.19}$$

□

REFERÊNCIAS

[1] M. Mitzenmacher. E. Upfal. *Probability and Computing. Randomization and Probabilistic Techniques in Algorithms and Data Analysis*, 2nd edn. Cambridge University Press. 2017.

[2] W. J. Stewart. *Probability, Markov Chains, Queues, and simulatons.* Princeton University Press, 2009.

[3] W. Krauth. *Statistical Mechanics Algorithms and Computations*, OUP Oxford. 2006.

[4] J. Dongarra F. Sullivan. *Guest Editors' Introduction: The Top 10 Algorithms.* Computing in Science and Engineering, 2000.

[5] M. Jerrum A. Sinclair. *The Markov chain Monte Carlo method: an approach to approximate counting and integration.* Approximation algorithms for NP-hard problems (pp. 482–519). PWS Publishing. 1996.

[6] C. Andrieu, N. DE Freitas, A. Doucet M. I. Jordan. *An Introduction to MCMC for Machine Learning.* MachineLearning, 50:5–43. 2003.

[7] C. Moore S. Mertens. *The Nature of Computation*, OUP Oxford. 2011

[8] Frobenius, G. *Über matrizen aus positiven elementen.* B. Preuss. Akad. Wiss. Berlin, Germany. 1908.

[9] Perron, O. *Zur theorie der matrices.* Mathematische Annalen, 64 (2), 248–263. 1907.

[10] S.K. Berberian. *Linear Algebra.* Oxford University Press. 1992.

[11] W. L. Dunn, J. K. Shultis. *Exploring Monte Carlo Methods* Elsevier Science, 2011.

[12] M.R. Jerrum, L.G. Valiant, e V.V. Vazirani. *Random generation of combinatorial structures from a uniform distribution* Theoretical Computer Science, 43:169–188, 1986.

[13] D. A. Levin, Y. Peres E. L. Wilmer. *Markov Chains and Mixing Times.* 2nd edn American Mathematical Society, 2017.

[14] F. Chung. *Four proofs for the Cheeger inequality and graph partition algorithms.* ICCM. 2007.

[15] M. Newman. *Networks* OUP Oxford; 2nd edn, 2018.

[16] S. Oldham, B. Fulcher, L. Parkes, A. Arnatkeviciute, C. Suo, A. Fornito *Consistency and differences between centrality measures across distinct classes of networks* PLoS ONE 14(7): e0220061, 2019.

[17] L. C. Freeman *A set of measures of centrality based upon betweenness* Sociometry. 40 (1): 35–41. 1977.

[18] M. H. Chehreghani, T. Abdessalem, A. Bifet. *Metropolis-Hastings Algorithms for Estimating Betweenness Centrality* EDBT, 2019.

[19] A. Marsden. *Eigenvalues of the Laplacian and Their Relationship to the Connectedness of a Graph* University of Chicago, REU, 2013.

[20] J. Cheeger. *A lower bound for the smallest eigenvalue of the Laplacian* Problems in analysis (Papers dedicated to Salomon Bochner, 1969). Princeton, N. J.: Princeton Univ. Press. pp. 195–199.

[21] Orponen, P., and S. E. Schaeffer. *On the NP-Completeness of Some Graph ClusterMeasures* Proceedingsof the 4th International Workshop on Efficient and Ex-perimental Algorithms (WEA'05, Santorini, Greece, May 2005), edited by S. Nikoletseas, volume 3503 of Lect. Notes Comp. Sci., pp.524–533. 2005.