Universidade Federal do Rio de Janeiro

Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa de Engenharia

Programa de Engenharia de Sistemas e Computação

CPS767 - Algoritmos de Monte Carlo e Cadeias de Markov

Prof. Daniel Ratton Figueiredo

3ª Lista de Exercícios

Luiz Henrique Souza Caldas
email: lhscaldas@cos.ufrj.br

9 de maio de 2025

Questão 1: Passeios aleatórios enviesados

Considere um grafo não direcionado $G = (V, E)$ com peso nas arestas, tal que $w_{ij} > 0$ para toda aresta $(i, j) \in E$. Considere um andarilho aleatório que caminha por este grafo em tempo discreto, mas cujos passos são enviesados pelos pesos das arestas. Em particular, a probabilidade do andarilho ir do vértice i para o vértice j é dada por w_{ij}/W_i, onde $W_i = \sum_j w_{ij}$ (soma dos pesos das arestas incidentes ao vértice $i \in V$). Temos assim um passeio aleatório enviesado linearmente pelos pesos das arestas.

1. Mostre que este passeio aleatório induz uma cadeia de Markov calculando a matriz de transição de probabilidade.

> Conforme o enunciado, temos que a probabilidade de transição do estado i para o estado j é dada por:
>
> $$P_{ij} = \frac{w_{ij}}{W_i},$$
>
> onde $W_i = \sum_j w_{ij}$ é a soma dos pesos das arestas incidentes ao vértice i e $w_{ij} > 0$ é o peso da aresta que liga os vértices i e j.
>
> Como $w_{ij} > 0$ para todas as arestas $(i, j) \in E$ e $W_i > 0$, temos que $P_{ij} \geq 0$, sendo estritamente positivo sempre que houver uma aresta entre i e j.
>
> Além disso,
>
> $$\sum_j P_{ij} = \sum_j \frac{w_{ij}}{W_i} = \frac{1}{W_i} \sum_j w_{ij} = \frac{W_i}{W_i} = 1$$
>
> Assim, temos que a soma das probabilidades de transição do estado i para todos os outros estados j é igual a 1, o que caracteriza uma matriz de transição de probabilidade.
>
> Portanto, o passeio aleatório enviesado <u>induz uma cadeia de Markov</u>.

2. Determine a distribuição estacionária desta cadeia de Markov (dica: use o método da inspeção).

> Uma distribuição estacionária π é uma distribuição de probabilidade que satisfaz a seguinte equação:
>
> $$\pi P = \pi$$
>
> Para cada estado i da distribuição estacionária, temos que:
>
> $$\pi_i = \sum_j \pi_j P_{ji}$$

Usando o resultado do item anterior, temos que:

$$\pi_i = \sum_j \pi_j \frac{w_{ji}}{W_j}$$

Supondo que a distribuição estacionária é proporcional à soma do peso das arestas incidentes ao vértice j, ou seja, $\pi_j = ZW_j$, onde $Z > 0$ é uma constante, temos que:

$$\pi_i = \sum_j (ZW_j) \frac{w_{ji}}{W_j} = Z \sum_j w_{ji}$$

Como o grafo é não direcionado, temos que $w_{ij} = w_{ji}$, ou seja:

$$\pi_i = Z \sum_j w_{ji} = Z \sum_j w_{ij} = ZW_i$$

Porém, sabemos que $\sum_k \pi_k = 1$, ou seja:

$$\sum_k \pi_k = \sum_k ZW_k = 1 \quad \Rightarrow \quad Z = \frac{1}{\sum_k W_k}$$

Portanto, a distribuição estacionária é dada por:

$$\boxed{\pi_i = \frac{W_i}{\sum_k W_k}}$$

3. Determine se esta cadeia de Markov é reversível no tempo.

Uma Cadeia de Markov é dita reversível no tempo se, e somente se, satizfaz a equação de fluxo de massa de probabilidade:

$$\pi_i P_{ij} = \pi_j P_{ji}$$

Aplicando a distribuição estacionária que encontramos no item anterior, temos que:

$$\pi_i P_{ij} = \frac{W_i}{\sum_k W_k} \cdot \frac{w_{ij}}{W_i} = \frac{w_{ij}}{\sum_k W_k} \quad \text{e} \quad \pi_j P_{ji} = \frac{W_j}{\sum_k W_k} \cdot \frac{w_{ji}}{W_j} = \frac{w_{ji}}{\sum_k W_k}$$

Como $w_{ij} = w_{ji}$, temos que:

$$\frac{w_{ij}}{\sum_k W_k} = \frac{w_{ji}}{\sum_k W_k} \quad \Rightarrow \quad \boxed{\pi_i P_{ij} = \pi_j P_{ji}}$$

Portanto, <u>a cadeia de Markov é reversível</u> no tempo.

Questão 2: Convergência de passeios aleatórios

Considere um passeio aleatório preguiçoso (com $p = 1/2$) caminhando sobre um grafo com n vértices. Estamos interessados em entender a convergência da distribuição $\pi(t)$ em diferentes grafos. Assuma que o passeio sempre inicia sua caminhada no vértice 1, ou seja, $\pi_1(0) = 1$. Considere os seguintes grafos: grafo em anel ($n = 125$), árvore binária cheia ($n = 127$), grafo em reticulado (grid) com duas dimensões ($n = 121$).

1. Para cada grafo, construa a matriz de transição de probabilidade (ou seja, determine P_{ij} para todo vértice i, j do grafo). Atenção com a numeração dos vértices!

Independente do grafo, a matriz de transição de probabilidade P é dada por:

$$P_{ij} = \begin{cases} 1/2 & \text{se } i = j \\ 1/2d_i & \text{se } [i \neq j] \wedge [(i,j) \in E] \\ 0 & \text{caso contrário} \end{cases}$$

onde d_i é o grau do vértice i e E é o conjunto de arestas do grafo, ou seja, pares (i, j) que representam arestas entre os vértices i e j, se a aresta existir.

- Grafo em anel ($n = 125$):

$$d_i = 2 \quad \forall i \in \{1, \dots, 125\}$$

$$P_{ij} = \begin{cases} 1/2 & \text{se } i = j \\ 1/4 & \text{se } [i \neq j] \wedge [(i,j) \in E] \\ 0 & \text{caso contrário} \end{cases}$$

- Grafo em árvore binária cheia ($n = 127$):
  Começando a numeração do nó raiz como 1, temos:

$$2i \leq 127 \quad \Rightarrow \quad i \leq 63.5$$

  Isso significa que os nós $i \in \{2, \dots, 63\}$ são nós internos, enquanto os nós $i \in [64, 127]$ são folhas. Assim, temos:

$$d_i = \begin{cases} 2 & \text{se } i \text{ é raiz } (i = 1) \\ 3 & \text{se } i \text{ é nó interno } (i \in \{2, \dots, 63\}) \\ 1 & \text{se } i \text{ é folha } (i \in \{64, \dots, 127\}) \end{cases}$$

$$P_{ij} = \begin{cases} 1/2 & \text{se } i = j \\ 1/4 & \text{se } [i \neq j] \wedge [(i,j) \in E] \wedge [i = 1] \quad \text{(raiz)} \\ 1/6 & \text{se } [i \neq j] \wedge [(i,j) \in E] \wedge [i \in \{2, \dots, 63\}] \quad \text{(nós internos)} \\ 1/2 & \text{se } [i \neq j] \wedge [(i,j) \in E] \wedge [i \in \{64, \dots, 127\}] \quad \text{(folhas)} \\ 0 & \text{caso contrário} \end{cases}$$

- Grafo em reticulado (grid) com duas dimensões ($n = 121$):
  Numerando os vétices de forma crescente da esquerda para a direita e de cima para baixo, temos:

$$d_i = \begin{cases} 2 & \text{se } i \in \{1, 11, 111, 121\} \quad \text{(cantos)} \\ 3 & \text{se } \begin{cases} i \in \{2, \dots, 10\} \quad \text{(borda de cima)} \\ i \in \{112, \dots, 120\} \quad \text{(borda de baixo)} \\ i \in \{12, 23, 34, 45, 56, 67, 78, 89, 100\} \quad \text{(borda da esquerda)} \\ i \in \{22, 33, 44, 55, 66, 77, 88, 99, 110\} \quad \text{(borda da direita)} \end{cases} \\ 4 & \text{caso contrário (nós internos)} \end{cases}$$

$$P_{ij} = \begin{cases} 1/2 & \text{se } i = j \\ 1/4 & \text{se } i \neq j,\ (i,j) \in E \text{ e } i \text{ é canto} \\ 1/6 & \text{se } i \neq j,\ (i,j) \in E \text{ e } i \text{ é borda} \\ 1/8 & \text{se } i \neq j,\ (i,j) \in E \text{ e } i \text{ é nó interno} \\ 0 & \text{caso contrário} \end{cases}$$

2. Determine analiticamente a distribuição estacionária para cada grafo (ou seja, determine π_i para cada vértice i do grafo).

Independente do grafo (dentre os 3 do enunciado), a matriz de transição de probabilidade P do passeio preguiçoso é dada por:

$$P = \frac{1}{2}I + \frac{1}{2}P',$$

onde o termo $\frac{1}{2}I$ representa a probabilidade de permanecer no mesmo vértice, e o termo $\frac{1}{2}P'$ representa a probabilidade de transitar para um vértice vizinho.

A equação de equilíbrio estacionário fica:

$$\pi = \pi P = \pi \left(\frac{1}{2}I + \frac{1}{2}P' \right) = \frac{1}{2}\pi + \frac{1}{2}\pi P'.$$

Subtraindo $\frac{1}{2}\pi$ de ambos os lados:

$$\frac{1}{2}\pi = \frac{1}{2}\pi P' \Rightarrow \pi = \pi P'.$$

Ou seja, o *self-loop* introduzido pelo passeio preguiçoso não altera a distribuição estacionária. Basta, portanto, calcular a distribuição estacionária do passeio aleatório padrão, com matriz de transição P'.

Para o passeio aleatório padrão, a distribuição estacionária é dada por:

$$\pi_i = \frac{d_i}{K}, \text{ onde } K = \sum_i d_i = 2m, \text{ sendo } m \text{ o número de arestas do grafo.}$$

- Grafo em anel ($n = 125$):

$$d_i = 2 \quad \forall i \in \{1, \dots, 125\}$$

$$m = 125$$

$$\pi_i = \frac{2}{2 \cdot 125} = \frac{1}{125} \quad \forall i \in \{1, \dots, 125\}$$

- Grafo em árvore binária cheia ($n = 127$):

$$d_i = \begin{cases} 2 & \text{se } i \text{ é raiz } (i = 1) \\ 3 & \text{se } i \text{ é nó interno } (i \in \{2, \dots, 63\}) \\ 1 & \text{se } i \text{ é folha } (i \in \{64, \dots, 127\}) \end{cases}$$

$$m = 127 - 1 = 126$$

$$\pi_i = \begin{cases} \frac{2}{2 \cdot 126} = \frac{1}{126} & \text{se } i \text{ é raiz } (i = 1) \\ \frac{3}{2 \cdot 126} = \frac{1}{84} & \text{se } i \text{ é nó interno } (i \in \{2, \dots, 63\}) \\ \frac{1}{2 \cdot 126} = \frac{1}{252} & \text{se } i \text{ é folha } (i \in \{64, \dots, 127\}) \end{cases}$$

- Grafo em reticulado (grid) com duas dimensões ($n = 121$):

$$d_i = \begin{cases} 2 & \text{se } i \in \{1, 11, 111, 121\} \quad \text{(cantos)} \\ 3 & \text{se } \begin{cases} i \in \{2, \dots, 10\} & \text{(borda de cima)} \\ i \in \{112, \dots, 120\} & \text{(borda de baixo)} \\ i \in \{12, 23, 34, 45, 56, 67, 78, 89, 100\} & \text{(borda da esquerda)} \\ i \in \{22, 33, 44, 55, 66, 77, 88, 99, 110\} & \text{(borda da direita)} \end{cases} \\ 4 & \text{caso contrário (nós internos)} \end{cases}$$

$$m_1 = 10 \cdot 11 = 110 \quad \text{(conexões verticais)}$$

$$m_2 = 10 \cdot 11 = 110 \quad \text{(conexões horizontais)}$$

$$m = m_1 + m_2 = 110 + 110 = 220$$

$$\pi_i = \begin{cases} \frac{2}{2\cdot220} = \frac{1}{220} & \text{se } i \in \{1, 11, 111, 121\} \quad \text{(cantos)} \\ \frac{3}{2\cdot220} = \frac{3}{440} & \text{se } \begin{cases} i \in \{2, \ldots, 10\} & \text{(borda de cima)} \\ i \in \{112, \ldots, 120\} & \text{(borda de baixo)} \\ i \in \{12, 23, 34, 45, 56, 67, 78, 89, 100\} & \text{(borda da esquerda)} \\ i \in \{22, 33, 44, 55, 66, 77, 88, 99, 110\} & \text{(borda da direita)} \end{cases} \\ \frac{4}{2\cdot220} = \frac{1}{110} & \text{caso contrário (nós internos)} \end{cases}$$

3. Para cada grafo, calcule numericamente a variação total entre $\pi(t)$ e a distribuição estacionária, para $t = 0, 1, \ldots$. Trace um gráfico onde cada curva corresponde a um grafo (preferencialmente em escala log-log, com $t \in [1, 10^3]$).

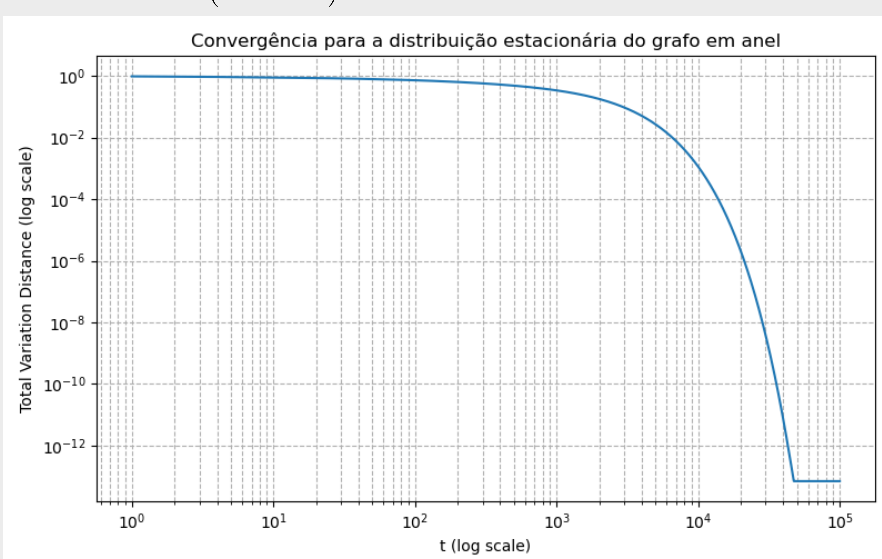
Foi implementado um código em Python para gerar as matrizes de transição de probabilidade P e as distribuições estacionárias π e calcular a variação total entre $\pi(t)$ e a distribuição estacionária, utilizando a formula:

$$\text{Variação Total} = \sum_i |\pi_i(t) - \pi_i|$$

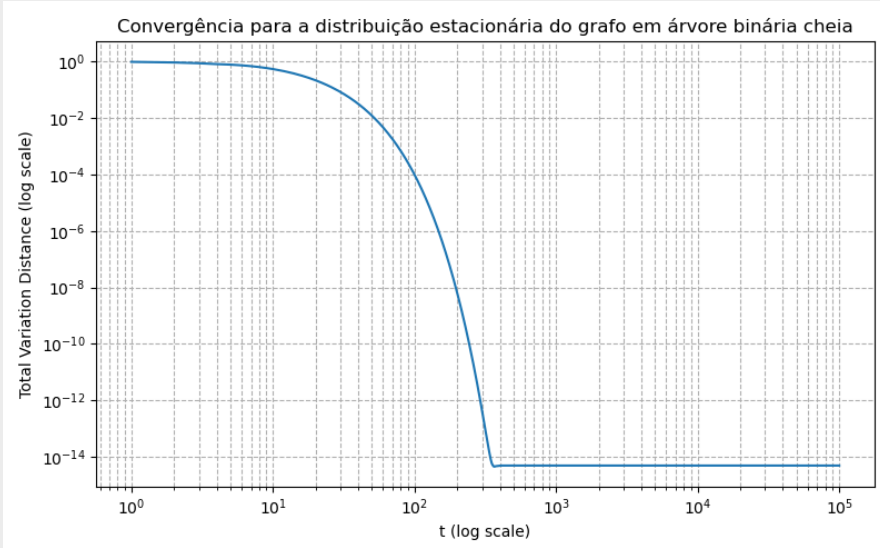
O link para o código está disponível no repositório do GitHub citado no fim deste relatório.

Para cada grafo, foi considerada a distribuição inicial $\pi_1(1) = 1$ e, apesar do enunciado sugerir 10^3 iterações, foram utilizadas 10^5 para vizualizar melhor a convergência. Todos os graficos estão em escala log-log.

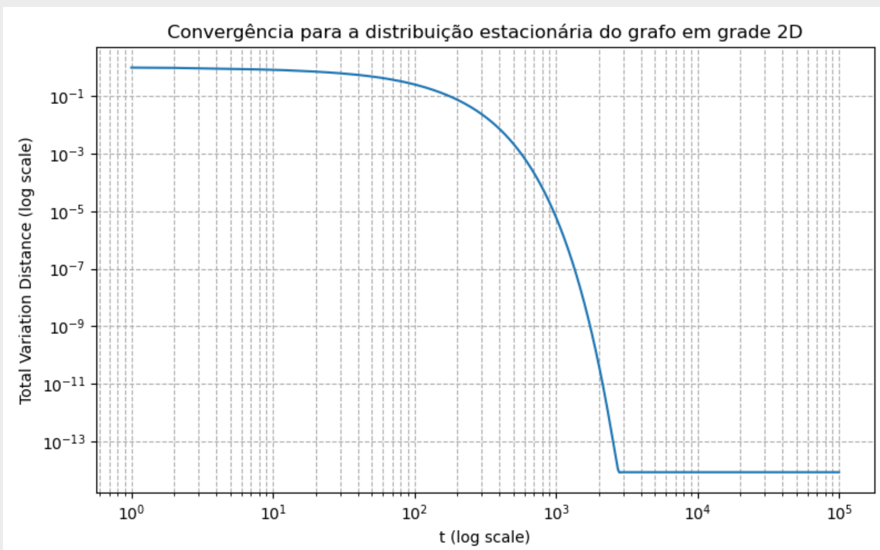
- Grafo em anel ($n = 125$):



- Grafo em árvore binária cheia ($n = 127$):



- Grafo em reticulado (grid) com duas dimensões ($n = 121$):



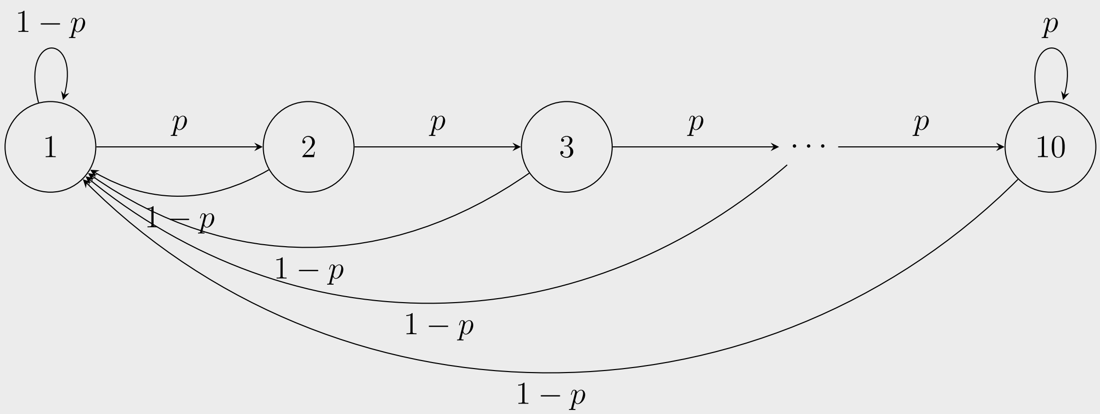
4. O que você pode concluir sobre a convergência em função da estrutura do grafo?

O grafo em árvore binária cheia é o que apresenta convergência mais rápida, seguido pelo grafo em reticulado (grid) e, por último, o grafo em anel. Isso pode ser explicado pela estrutura de cada grafo: a árvore possui uma organização ramificada que permite ao passeio aleatório atingir diferentes regiões rapidamente. O reticulado também possui múltiplos caminhos entre os vértices, facilitando a dispersão da probabilidade ao longo do tempo. Já o grafo em anel, com conexões mais limitadas e estrutura cíclica, dificulta essa dispersão e, portanto, tende a convergir mais lentamente para a distribuição estacionária.

Questão 3: Tempo de mistura

Considere um processo estocástico que inicia no estado 1 e a cada instante de tempo incrementa o valor do estado em uma unidade com probabilidade p ou retorna ao estado inicial (estado 1) com probabilidade $1 - p$. No estado n o processo não cresce mais, e se mantém neste estado com probabilidade p. Assuma que $n = 10$ e que $p \in \{0.25, 0.5, 0.75\}$.

1. Construa a cadeia de Markov deste processo mostrando a matriz de transição de probabilidade em função de p.



$$P_{ij} = \begin{cases} 1-p & \text{se } j = 1 \\ p & \text{se } (j = i+1) \vee (i = j = 10) \\ 0 & \text{caso contrário} \end{cases}$$

Matriz de transição (parcial):

$$\begin{bmatrix} (1-p) & p & 0 & \cdots & 0 \\ (1-p) & 0 & p & & 0 \\ \vdots & & & \ddots & \vdots \\ (1-p) & 0 & 0 & \cdots & p \\ (1-p) & 0 & 0 & \cdots & p \end{bmatrix}$$

2. Determine numericamente o vão espectral da cadeia de Markov para cada valor de p.

O vão espectral é dado por:

$$\delta = 1 - |\lambda_2|$$

onde λ_2 é o segundo maior autovalor (em módulo) da matriz de transição P.

Utilizando um código em Python, com a biblioteca NumPy, podemos calcular os autovalores da matriz de transição P para cada valor de p. O código pode ser visto no repositório do Github cujo link encontra-se no fim deste relatório.

- **Para** $p = 0{,}25$**:**
  - Matriz de transição (parcial): $\begin{bmatrix} 0.75 & 0.25 & 0 & \cdots & 0 \\ 0.75 & 0 & 0.25 & & 0 \\ \vdots & & & \ddots & \vdots \\ 0.75 & 0 & 0 & \cdots & 0.25 \\ 0.75 & 0 & 0 & \cdots & 0.25 \end{bmatrix}$
  - Maiores autovalores em módulo: $|\lambda_1| = 1.0, \quad |\lambda_2| \approx 1.7125 \times 10^{-4}$
  - Vão espectral: $\boxed{\delta = 0.9998287}$
- **Para** $p = 0{,}5$**:**
  - Matriz de transição (parcial): $\begin{bmatrix} 0.5 & 0.5 & 0 & \cdots & 0 \\ 0.5 & 0 & 0.5 & & 0 \\ \vdots & & & \ddots & \vdots \\ 0.5 & 0 & 0 & \cdots & 0.5 \\ 0.5 & 0 & 0 & \cdots & 0.5 \end{bmatrix}$
  - Maiores autovalores em módulo: $|\lambda_1| = 1.0, \quad |\lambda_2| \approx 3.4251 \times 10^{-4}$
  - Vão espectral: $\boxed{\delta = 0.9996575}$
- **Para** $p = 0{,}75$**:**
  - Matriz de transição (parcial): $\begin{bmatrix} 0.25 & 0.75 & 0 & \cdots & 0 \\ 0.25 & 0 & 0.75 & & 0 \\ \vdots & & & \ddots & \vdots \\ 0.25 & 0 & 0 & \cdots & 0.75 \\ 0.25 & 0 & 0 & \cdots & 0.75 \end{bmatrix}$
  - Maiores autovalores em módulo: $|\lambda_1| = 1.0, \quad |\lambda_2| \approx 4.3546 \times 10^{-6}$
  - Vão espectral: $\boxed{\delta = 0.9999956}$

3. Determine numericamente a distribuição estacionária para cada valor de p, e indique o estado de menor probabilidade.

O estado estacionário π de uma cadeia de Markov é um vetor de probabilidade tal que:

$$\pi P = \pi$$

Ou seja, ele é um autovetor à esquerda da matriz de transição P associado ao autovalor $\lambda = 1$. A distribuição estacionária pode ser obtida utilizando novamente um código Python com a biblioteca NumPy para computar os autovalores e autovetores de $P^\top$, selecionando aquele correspondente a $\lambda = 1$ e normalizando-o para que $\sum_i \pi_i = 1$.

- **Para** $p = 0{,}25$**:**
  - Distribuição estacionária: $\pi = \begin{bmatrix} 0.75 \\ 0.1875 \\ 0.046875 \\ 0.01171875 \\ 0.00292969 \\ 0.00073242 \\ 0.00018311 \\ 0.00004578 \\ 0.00001144 \\ \boxed{0.00000381} \end{bmatrix}$
  - Estado de menor probabilidade: **10**, com probabilidade $\boxed{\pi_o = 0.00000381}$.
- **Para** $p = 0{,}5$**:**
  - Distribuição estacionária: $\pi = \begin{bmatrix} 0.5 \\ 0.25 \\ 0.125 \\ 0.0625 \\ 0.03125 \\ 0.015625 \\ 0.0078125 \\ 0.00390625 \\ \boxed{0.00195312} \\ \boxed{0.00195312} \end{bmatrix}$
  - Estado de menor probabilidade: **9** e **10**, com probabilidade $\boxed{\pi_o = 0.00195312}$.
- **Para** $p = 0{,}75$**:**
  - Distribuição estacionária: $\pi = \begin{bmatrix} 0.25 \\ 0.1875 \\ 0.140625 \\ 0.10546875 \\ 0.07910156 \\ 0.05932617 \\ 0.04449463 \\ 0.03337097 \\ \boxed{0.02502823} \\ 0.07508469 \end{bmatrix}$
  - Estado de menor probabilidade: **9**, com probabilidade $\boxed{\pi_o = 0.02502823}$.

4. Utilizando os dados acima, determine um limitante inferior e superior para o tempo de mistura quando $\epsilon = 10^{-6}$ para cada valor de p.

O tempo de mistura τ_ϵ representa, de forma aproximada, o número de passos necessários para que a distribuição da cadeia de Markov esteja a uma distância total menor que ϵ da distribuição estacionária, independentemente do estado inicial.

Ele pode ser estimado por:

$$\left(\frac{1}{\delta} - 1\right) \log \frac{1}{2\epsilon} \leq \tau_\epsilon \leq \frac{1}{\delta} \log \frac{1}{\pi_o \epsilon},$$

onde δ é o vão espectral da matriz de transição, e $\pi_o = \min_i \pi_i$ é a menor entrada da distribuição estacionária.

- **Para** $p = 0{,}25$**:**
  - $\delta = 0.9998287$ e $\pi_o = 0.00000381$.
  - Limite inferior estimado: $\boxed{\tau_\epsilon \geq 0.0022476286040300246}$
  - Limite superior estimado: $\boxed{\tau_\epsilon \leq 26.296663189659515}$
- **Para** $p = 0{,}5$**:**
  - $\delta = 0.9996575$ e $\pi_o = 0.00195312$.
  - Limite inferior estimado: $\boxed{\tau_\epsilon \geq 0.004496027298161922}$
  - Limite superior estimado: $\boxed{\tau_\epsilon \leq 20.060706093968722}$
- **Para** $p = 0{,}75$**:**
  - $\delta = 0.9999956$ e $\pi_o = 0.02502823$.
  - Limite inferior estimado: $\boxed{\tau_\epsilon \geq 5.7142421447070306 \times 10^{-5}}$
  - Limite superior estimado: $\boxed{\tau_\epsilon \leq 17.50333771810467}$

5. O que você pode concluir sobre a influência de p no tempo de mistura?

O aumento de p tende a reduzir o tempo de mistura, pois favorece o avanço entre os estados e acelera a aproximação da distribuição estacionária. Já valores menores de p provocam retornos frequentes ao estado inicial, atrasando a convergência.

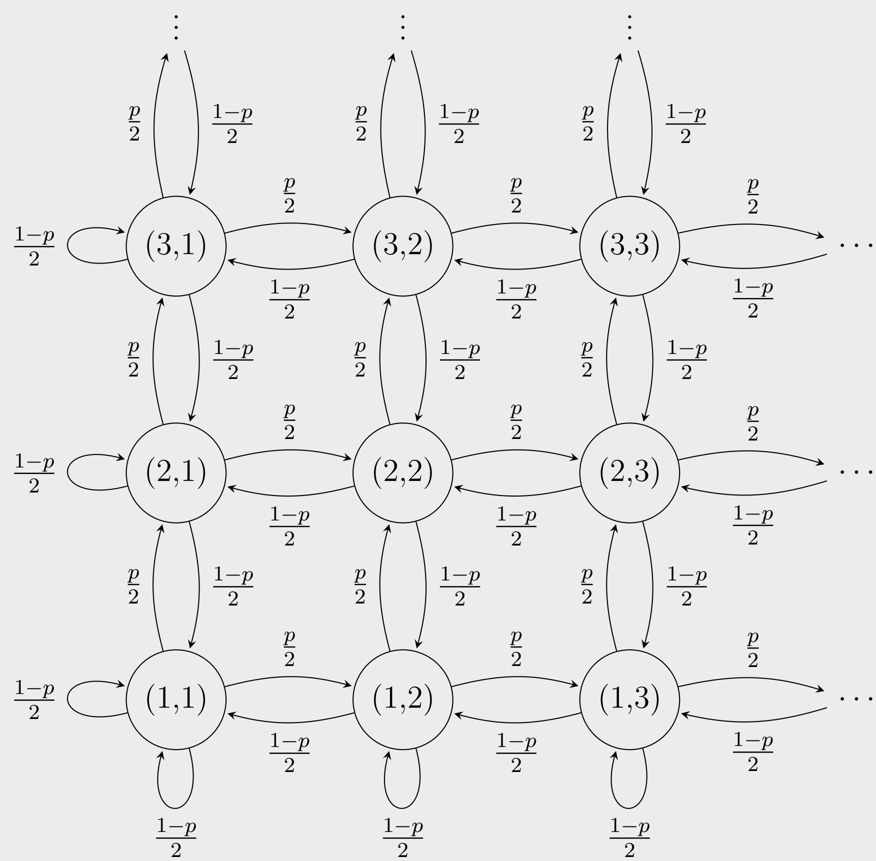
Os limites do tempo de mistura refletem esse efeito de formas distintas. O superior diminui com p, pois depende inversamente do vão espectral e da menor entrada da distribuição estacionária, ambos crescentes. O inferior, porém, pode crescer inicialmente e depois cair, já que envolve o termo $\left(\frac{1}{\delta} - 1\right)$, que varia de forma não linear com p.

Questão 4: Voltando à origem

Considere uma cadeia de Markov cujo espaço de estados é um láttice de duas dimensões sobre os números naturais, ou seja, $S = \{(i, j) \mid i \geq 1, j \geq 1\}$. Cada estado pode transicionar para um de seus vizinhos no láttice. Entretanto, se afastar da origem (se movimentar para o norte ou para o leste) tem probabilidade $p/2$, e se aproximar da origem tem probabilidade $(1 - p)/2$, onde p é um parâmetro do modelo (nas bordas, utilize self-loops). Assuma que $p \in \{0.25, 0.35, 0.45\}$.

1. Construa um simulador para essa cadeia de Markov.

A representação da Cadeia de Markov pode ser vista na figura abaixo:

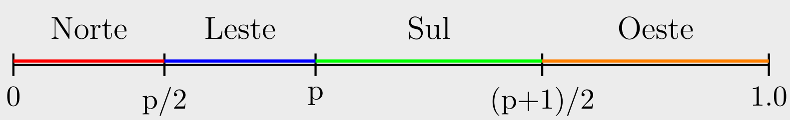


Os elementos da Matriz de Transição P podem ser calculados da seguinte forma:

$$P_{(i,j),(i',j')} = \begin{cases} \frac{p}{2} & \text{se } i' = i+1 \land j' = j \quad \text{(norte)} \\ \frac{p}{2} & \text{se } i' = i \land j' = j+1 \quad \text{(leste)} \\ \frac{1-p}{2} & \text{se } i > 1 \land i' = i-1 \land j' = j \quad \text{(sul)} \\ \frac{1-p}{2} & \text{se } j > 1 \land i' = i \land j' = j-1 \quad \text{(oeste)} \\ \frac{1-p}{2} & \text{se } i = 1 \land i' = i \land j' = j \quad \text{(borda inferior)} \\ \frac{1-p}{2} & \text{se } j = 1 \land i' = i \land j' = j \quad \text{(borda esquerda)} \\ 0 & \text{caso contrário} \end{cases}$$

Como a cadeia de Markov possui um número infinito de estados, a matriz de transição completa não pode ser armazenada integralmente na memória do computador. No entanto, para cada estado, conhecemos seus vizinhos e as respectivas probabilidades de transição. Assim, é possível implementar um código que, com base no estado atual, gera o próximo estado como um "dado enviesado", ou seja, uma amostra aleatória com probabilidades associadas a cada vizinho.

Ou seja, a cada iteração, o simulador deverá, com base no estado atual (i, j), escolher um dos vizinhos (i', j') com probabilidade $P_{(i,j),(i',j')}$. Para isso, o código pode gerar um número aleatório entre 0 e 1 e compará-lo com a distribuição acumulada das probabilidades dos vizinhos. O próximo estado será aquele cuja faixa acumulada contém o número gerado, simulando assim um dado enviesado de acordo com as probabilidades de transição.



2. Utilize o simulador para estimar a distribuição estacionária da origem (estado $(1, 1)$), ou seja $\pi_{1,1}$, para cada valor de p. Dica: utilize os tempos de retorno!

Pelo Teorema Ergódico forte, se a Cadeia de Markov é irredutível, aperiódica e com distribuição estacionária π, temos:

$$\lim_{k\to\infty} \frac{1}{k} \sum_{t=0}^{k-1} f(X_t) = E_\pi[f(X_t)],$$

onde $f(X_t)$ é uma função qualquer de X_t e $E_\pi[f(X_t)]$ é o valor esperado de $f(X_t)$ sob a distribuição estacionária π.

Fazendo $f(X_t) = I(X_t)$, uma função indicadora que retorna 1 se $X_t = (1, 1)$ e $f(X_t) = 0$ caso contrário, temos que:

$$\lim_{k\to\infty} \frac{1}{k}\sum_{t=0}^{k-1} I(X_t) = E_\pi[I(X_t)] = \pi_{1,1} \times 1 + (1-\pi_{1,1}) \times 0 = \pi_{1,1}$$

Com isso, podemos estimar a distribuição estacionária $\pi_{1,1}$ apenas simulando a cadeia de Markov por um número de passos suficientemente grande, contando quantas vezes o estado $(1,1)$ foi visitado e dividindo pelo número total de passos do caminho amostral. Assim, para 10^7 passos, temos:

- Para $p = 0{,}25$, a estimativa de $\pi_{(1,1)}$ foi: $\boxed{\pi_{(1,1)} \approx 0{,}4447}$
- Para $p = 0{,}35$, a estimativa de $\pi_{(1,1)}$ foi: $\boxed{\pi_{(1,1)} \approx 0{,}2137}$
- Para $p = 0{,}45$, a estimativa de $\pi_{(1,1)}$ foi: $\boxed{\pi_{(1,1)} \approx 0{,}0328}$

3. Seja $d(t)$ o valor esperado da distância (de Manhattan) entre X_t (o estado no tempo t) e a origem. Utilize o simulador para estimar $d(t)$ para $t \in \{10, 100, 1000\}$, para cada valor de p. O que você pode concluir?

A distância de Manhattan entre $X_t = (i_t, j_t)$ e a origem $(1,1)$ é dada por:

$$d_t = |i_t - 1| + |j_t - 1|$$

As estimativas de $d(t)$ para $t \in \{10, 100, 1000\}$, para cada valor de p podem ser obtidas simulando, para cada valor de p, a cadeia de Markov por 1000 passos e calculando a distância de Manhattan entre o estado atual e a origem. Como $d(t)$ é uma variável aleatória, podemos obter uma estimativa mais precisa rodando a simulação 30 vezes para cada valor de p e calculando a média das distâncias obtidas. Assim, temos:

p	$d(10)$	$d(100)$	$d(1000)$
0.25	2.70	3.10	3.00
0.35	3.53	4.57	4.00
0.45	4.50	8.97	11.40

OBS: os valores são decimais por ser uma média de 30 simulações.

Podemos concluir que, quanto maior o valor de p, mais os estados tendem a se afastar da origem ao longo do tempo. Isso ocorre porque a probabilidade de se mover para o norte ou para o leste aumenta com p, favorecendo direções que se afastam da origem, enquanto a probabilidade de se mover para o sul ou para o oeste, que aproximam o processo da origem, diminui.

Códigos

Os códigos utilizados para a resolução dos exercícios estão disponíveis no repositório do GitHub: `https://github.com/lhscaldas/CPS767_MCMC/`