

1 Filas e caixas: Considere um caixa de supermercado onde a cada instante de tempo uma pessoa chega probabilidade p , ou uma pessoa em atendimento sai do caixa com probabilidade q , ou nada acontece com probabilidade remanescente. Se uma pessoa chega e encontra o caixa vazio, ela imediatamente inicia o atendimento, caso contrário ela aguarda em uma fila. Ao terminar um atendimento, a próxima pessoa da fila imediatamente inicia o atendimento.

1.1 Construa uma cadeia de Markov para representar este sistema

Para modelar este problema como uma cadeia de Markov, podemos considerar os estados como o número de pessoas no caixa, incluindo a pessoa sendo atendida e as que estão em espera. Desta forma, o estado i representa o estado em que há i pessoas no caixa.

Assim, a cadeia pode ser modelada conforme a Figura 1.

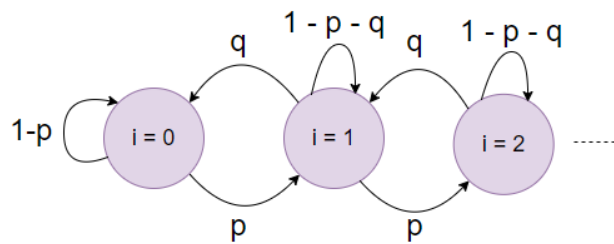


Figure 1: Representação da cadeia de Markov baseada nos parâmetros p e q .

Note que a cadeia pode ter infinitos estados pois o número de pessoas que chegam ao caixa é ilimitado no problema.

1.2 Encontre a distribuição estacionária assumindo que a fila possui capacidade infinita, e que $p < q$ (dica: use o método direto e monte uma recursão).

Considerando $\pi_k(S = i)$ a probabilidade em estarmos no estado i no tempo k , temos que, para $i > 0$:

$$\begin{aligned}\pi_k(i) &= P[S_k = i] \\ &= P[S_k = i | S_{k-1} = i-1] \pi_{k-1}(i-1) + P[S_k = i | S_{k-1} = i] \pi_{k-1}(i) + P[S_k = i | S_{k-1} = i+1] \pi_{k-1}(i+1) \\ &= p \pi_{k-1}(i-1) + (1-p-q) \pi_{k-1}(i) + q \pi_{k-1}(i+1)\end{aligned}$$

Na estacionariedade, $\pi_k(i) = \pi_{k-1}(i) = \pi(i)$. Portanto,

$$\pi(i) = p \pi(i-1) + (1-p-q) \pi(i) + q \pi(i+1)$$

Analogamente, para $i = 0$:

$$\pi(0) = q \pi(1) + (1-p) \pi(0)$$

Como $0 \leq p < q \rightarrow q > 0$, então:

$$\pi(1) = \frac{p}{q} \pi(0)$$

Utilizando estas equações, podemos perceber a seguinte recursão:

$$\pi(i+1) = \frac{p}{q} \pi(i)$$

O que, utilizando as condições iniciais, pode ser reformulada como:

$$\pi(i) = \left(\frac{p}{q}\right)^i \pi(0)$$

1.3 Determine a fração de tempo que o caixa passa ocioso (sem ter clientes para atender) em função dos parâmetros do modelo (dica: fração de tempo ocioso é dado pela probabilidade do sistema estar ocioso depois de um tempo muito grande).

Como $\pi(i)$ pode ser visto como a probabilidade de estarmos no estado i , então a fração de tempo que o caixa passa ocioso é representado por $\pi(0)$.

Uma vez que $\pi(i)$ representa uma probabilidade, então a soma das probabilidades para todos os estados deve ser 1. Matematicamente, temos:

$$\sum_{i=0}^{\infty} \pi(i) = 1$$

Aplicando a equação encontrada anteriormente, temos:

$$\begin{aligned} \sum_{i=0}^{\infty} \pi(i) &= \sum_{i=0}^{\infty} \left(\frac{p}{q}\right)^i \pi(0) \\ &= \pi(0) \sum_{i=0}^{\infty} \left(\frac{p}{q}\right)^i \\ &= \pi(0) \left(\frac{1}{1 - p/q}\right) \\ &= 1 \end{aligned}$$

Portanto, a fração de tempo que o caixa passa ocioso é:

$$\pi(0) = \frac{q - p}{p}$$

Note que, se $p = q$, então a taxa de pessoas que entram e saem do caixa é a mesma. Assim, o caixa nunca estará ocioso, o que é confirmado com $\pi(0) = 0$.

2 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 dado por w_{ij}/W_i , onde $W_i = \sum_j w_{ij}$ (W_i é a soma dos pesos das arestas incidentes ao vértice $i \in V$). Temos assim um passeio aleatório enviesado pelos pesos das arestas.

2.1 Mostre que este passeio aleatório induz uma cadeia de Markov.

Do enunciado, temos que a probabilidade de transição do vértice i para o vértice j é dada por:

$$P[S_k = j | S_{k-1} = i] = \frac{w_{ij}}{W_i}$$

Assim pela lei da probabilidade total:

$$\begin{aligned} \pi_k(j) &= P[S_k = j] \\ &= \sum_{i \in S} P[S_k = j | S_{k-1} = i] P[S_{k-1} = i] \\ &= \sum_{i \in S} \frac{w_{ij}}{W_i} \pi_{k-1}(i) \end{aligned}$$

Como $w_{ij} > 0$ para toda aresta $(i, j) \in E$, então o somatório pode ser simplificado para os vizinhos de j , pois $w_{ij} = 0$ para todos os vértices i não-vizinhos de j .

$$\pi_k(j) = \sum_{i \in N_j} \frac{w_{ij}}{W_i} \pi_{k-1}(i)$$

onde N_j é o conjunto de vértices vizinhos a j .

Pela equação acima, podemos ver que a probabilidade de estarmos no vértice j depende apenas da probabilidade em estarmos no vértice vizinho i . Portanto, a cadeia segue a propriedade *memoryless*, induzindo uma cadeia de Markov.

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

Na distribuição estacionária,

$$\pi_k(i) = \pi_{k-1}(i) = \pi(i)$$

Desta forma,

$$\pi(j) = \sum_{i \in N_j} \frac{w_{ij}}{W_i} \pi(i)$$

Supondo $\pi(j) = W_j / \sum_{i \in S} W_i$, então:

$$\begin{aligned} \pi(j) &= \sum_{i \in N_j} \frac{w_{ij}}{W_i} \cdot \frac{W_j}{\sum_{i \in S} W_i} \\ &= \frac{1}{\sum_{i \in S} W_i} \sum_{i \in N_j} w_{ij} \\ &= \frac{W_j}{\sum_{i \in S} W_i} \end{aligned}$$

Portanto, a distribuição estacionária é dada por:

$$\pi(j) = \frac{W_j}{\sum_{i \in S} W_i}$$

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

A cadeia é reversível no tempo se:

$$\pi(i)P[S_k = j | S_{k-1} = i] = \pi(j)P[S_k = i | S_{k-1} = j]$$

Assim,

$$\begin{aligned} \pi(j)P[S_k = i | S_{k-1} = j] &= \frac{W_j}{\sum_{i \in S} W_i} \cdot \frac{w_{ji}}{W_j} \\ &= \frac{w_{ji}}{\sum_{i \in S} W_i} \\ &= \frac{w_{ji}}{\sum_{i \in S} W_i} \cdot \frac{W_i}{W_i} \\ &= \pi(i) \cdot \frac{w_{ji}}{W_i} \end{aligned}$$

A cadeia é reversível no tempo somente se $w_{ij} = w_{ji}$, $\forall i, j \in S$.

3 Convergência de passeios aleatórios: Considere um passeio aleatório preguiçoso (com $p = 1/2$) caminhando sobre um grafo com $n = 100$ vértices. Estamos interessados em entender a convergência da distribuição $\pi(t)$ para 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, árvore binária cheia, grafo em reticulado com duas dimensões (grid 2D).

3.1 Para cada grafo, construa analiticamente a matriz de transição de probabilidade (ou seja, determine P_{ij} para todo vértice i, j do grafo).

Como o grafo é preguiçoso, em todas as iterações o passeio terá uma probabilidade $p = 1/2$ de permanecer no mesmo estado. Assim, $p_{ii} = 1/2$ para $i \in [1, 100]$. A probabilidade complementar será uniformemente distribuída entre os vértices vizinhos.

3.1.1 Grafo em Anel

Enumerando-se cada vértice do anel no sentido horário, como ilustrado na Figura 2, temos os seguintes valores de p_{ij} :

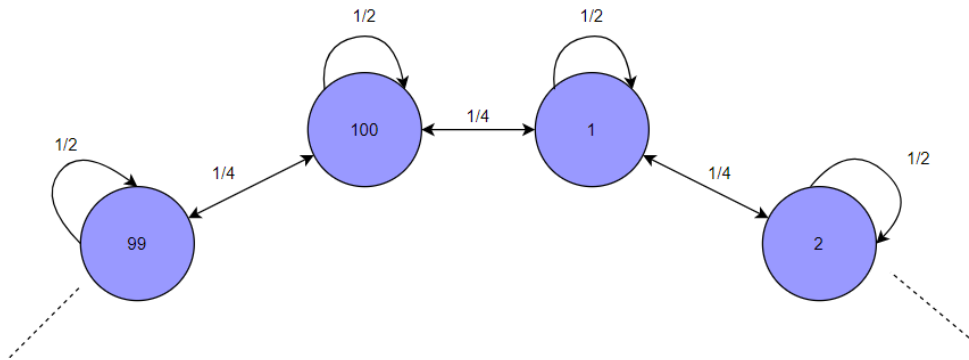


Figure 2: Probabilidades de transição do grafo em anel.

$$p_{ij} = \begin{cases} 1/4 & \text{se } |i - j| = 1 \text{ ou } (i = 100 \text{ e } j = 1) \text{ ou } (i = 1 \text{ e } j = 100) \\ 1/2 & \text{se } i = j \\ 0 & \text{caso contrário} \end{cases}$$

3.1.2 Árvore Binária Cheia

Para um árvore binária cheia de k níveis, temos $N = \sum_{i=0}^{k-1} 2^i$ vértices. No entanto, não há valor de k inteiro que faça $N = 100$. Portanto, arredondou-se o número de vértices para 63, fazendo com que a árvore tenha $k = 6$ níveis.

Os valores de p_{ij} da árvore binária cheia são representados nas Figuras 3, 4 e 5 (a enumeração foi feita partindo-se do nó raiz e, de nível em nível, aumenta-se a numeração para a direita). Assim, o nó raiz é representado por $i = 1$, os nós intermediários por $i \in [2, 31]$ e os nós-folha por $i \in [32, 63]$.

Analogamente ao grafo em anel, a probabilidade p_{ij} para $i \neq j$ é calculada distribuindo-se uniformemente a probabilidade complementar à p entre os vizinhos de i . Por exemplo, se i possui 3 vizinhos (1 pai e 2 filhos) e dado que a probabilidade de *self-loop* é $p = 1/2$, então $p_{ij} = (1 - p)/3 = 1/6$ de i para seus vizinhos (o que acontece quando $i \in [2, 31]$).

Note que apenas os nós-folha e o nó-raiz não terão 3 vizinhos. Em particular, o nó-raiz ($i = 1$) terá 2 vizinhos (seus únicos filhos) e, neste caso, $p_{1j} = 1/4$. Analogamente, $p_{ij} = 1/2$ quando i for um nó-raiz ($i \in [32, 63]$).

Esta construção faz com que a cadeia de Markov induzida seja direcionada. Isto é, p_{ij} não necessariamente é igual a p_{ji} , o que acontece para o nível 1 (nó-raiz) e nível 6 (nós-folha).

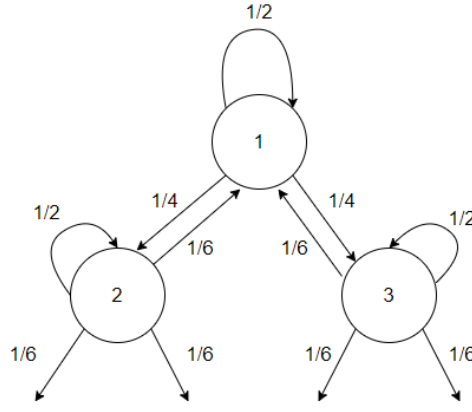


Figure 3: Representação da cadeia de Markov induzida no níveis 1 e 2 da árvore binária cheia, com o nó-raíz.

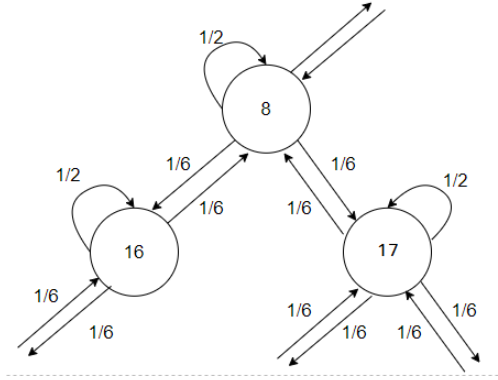


Figure 4: Representação da cadeia de Markov induzida nos níveis intermediários (2 ao 5) da árvore binária cheia.

3.1.3 Grafo em Reticulado

Para um grafo reticulado 10×10 , o passeio aleatório induz uma cadeia de Markov não-direcionada de forma que podemos fazer com que haja uma probabilidade não-nula do passeio ir de uma extremidade do reticulado à outra. Assim, dados i, j pontos nesse grid cujas coordenadas são dadas por (x_i, y_i) e (x_j, y_j) , então:

$$p_{ij} = \begin{cases} 1/2 & \text{se } x_i = x_j \text{ e } y_i = y_j \\ 1/8 & \text{se } y_j = y_i \pm 1 \text{ e } x_j = x_i, \text{ para } y_i \in [2, 9] \\ 1/8 & \text{se } x_j = x_i \pm 1 \text{ e } y_j = y_i, \text{ para } x_i \in [2, 9] \\ 1/8 & \text{se } x_i = 1 \text{ e } x_j = 10 \text{ e } y_j = y_i \\ 1/8 & \text{se } y_i = 1 \text{ e } y_j = 10 \text{ e } x_j = x_i \\ 0 & \text{caso contrário} \end{cases} \quad (1)$$

Uma representação dessas probabilidades pode ser vista da Figura 6.

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

3.2.1 Grafo em Anel

Pela simetria do grafo,

$$\pi_i = \pi_j, \forall i, j \in [1, 100]$$

Assim, como $\sum_{i=1}^{100} \pi_i = 1$, então

$$\pi_i = \frac{1}{100}$$

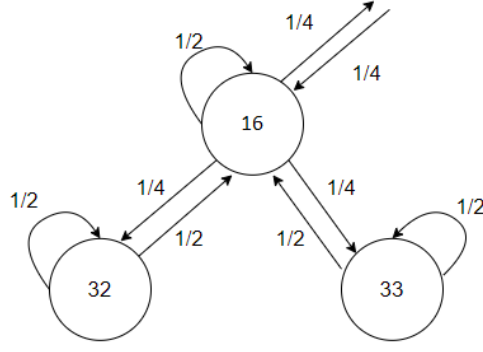


Figure 5: Representação da cadeia de Markov induzida no nível 6 da árvore binária cheia, com os nós-folha.

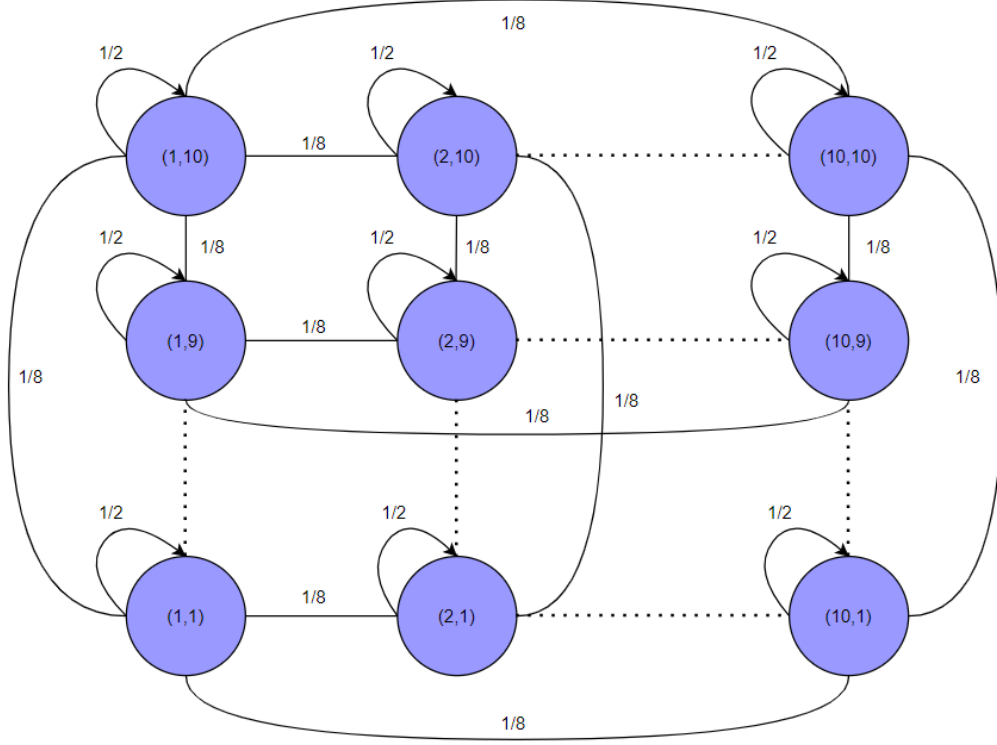


Figure 6: Representação da cadeia de Markov induzida no grafo em reticulado.

3.2.2 Árvore Binária

Uma vez que tenhamos os valores de p_{ij} , é possível montar a matriz de transição P . No entanto, sabemos que no estado estacionário,

$$\pi P = 1 \cdot \pi$$

Assim, a distribuição estacionária π é um autovetor à esquerda associado ao autovalor $\lambda = 1$ da matriz P . Portanto, basta que encontremos os autovetores à esquerda da matriz P .

O código-fonte para o cálculo desses autovetores é disponibilizado neste [github](#).

Em particular, vemos que os valores de π_i são uniformes para um mesmo nível da árvore binária cheia e o mesmo também não varia entre níveis intermediários. Assim, temos os seguintes valores de π_i :

$$\pi_i = \begin{cases} 0.01612903 & \text{se } i \text{ é o nó-raiz } (i = 1) \\ 0.02419355 & \text{se } i \text{ é um nó em nível intermediário } (i \in [2, 31]) \\ 0.00806452 & \text{se } i \text{ é um nó-folha } (i \in [32, 63]) \end{cases} \quad (2)$$

3.2.3 Grafo em Reticulado

Pela simetria do grafo,

$$\pi_i = \pi_j, \forall i, j \in [1, 100]$$

Assim, como $\sum_{i=1}^{100} \pi_i = 1$, então

$$\pi_i = \frac{1}{100}$$

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

Os resultados numéricos para os grafos em anel, árvore binária e reticulado estão apresentados nas Figuras 7, 8 e 9, respectivamente. Para cada gráfico, calculou-se também os valores da variação total d_{tv} para $p \in \{.1, .2, .3, .5, .8, .9\}$

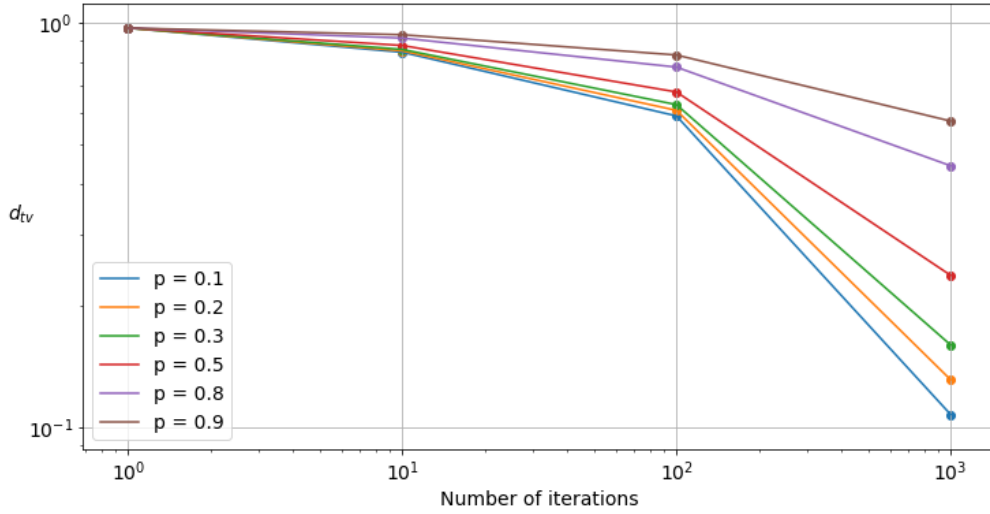


Figure 7: Variação total d_{tv} entre $\pi(t)$ e π para o grafo em anel em diferentes valores de p .

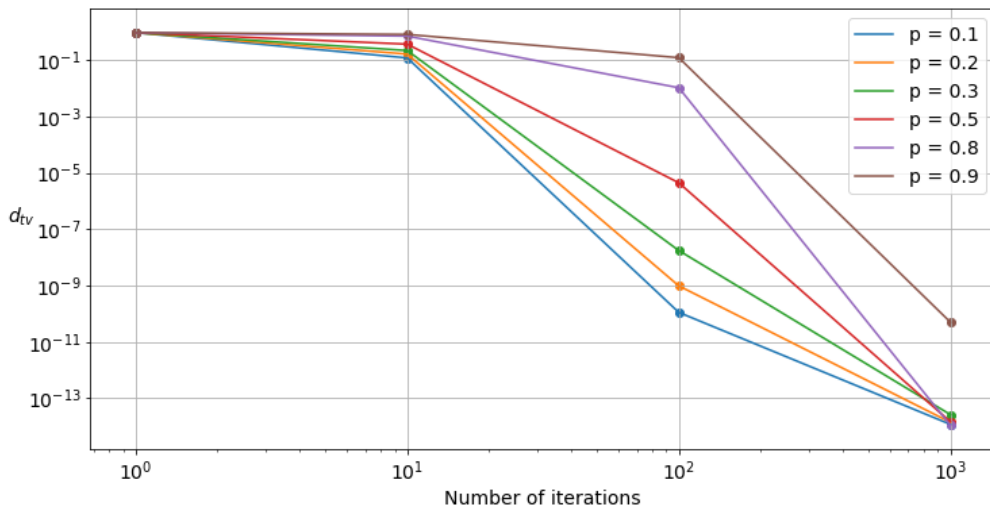


Figure 8: Variação total d_{tv} entre $\pi(t)$ e π para o grafo em árvore binária em diferentes valores de p .

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

De uma forma geral, o grafo em anel e em reticulado possuem as mesmas distribuições estacionárias para a mesma quantidade de estados. No entanto, o grau dos vértices no grafo reticulado é maior do que no grafo em

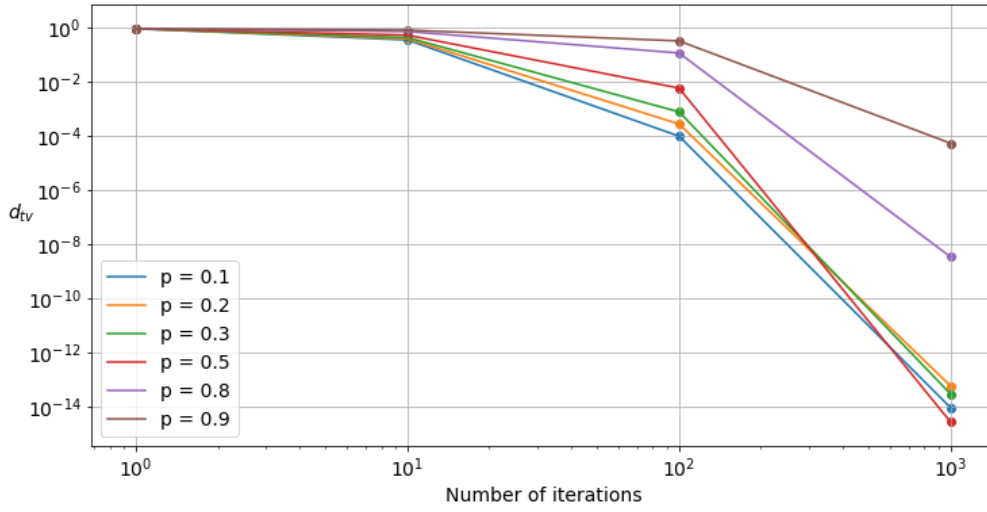


Figure 9: Variação total d_{tv} entre $\pi(t)$ e π para o grafo em reticulado em diferentes valores de p .

anel, o que faz com que o passeio possa atingir mais estados em um menor tempo. Assim, a convergência da variação total é consideravelmente mais rápida para o grafo reticulado.

Além disso, a variação do parâmetro p de *self-loops* mostra que, em geral, quanto menor o valor de p , mais rápida é a convergência de d_{tv} . Isso acontece intuitivamente pois, uma vez que há uma menor probabilidade do passeio ficar preso em um *self-loop*, mais rápido ele percorrerá todos os estados. Assim, para qualquer valor de $p < 1$ (ou seja, excluindo-se o caso em que o passeio fica preso no estado inicial), a convergência da variação total é atingida mesmo com os *self-loops*, que impactam somente na quantidade de iterações até a convergência.

4 Tempo de mistura em passeios aleatórios: Considere os passeios aleatórios preguiçosos da questão anterior, e assuma que $\epsilon = 10^{-4}$. Para cada grafo, determine o tempo de mistura, τ_n (com ϵ fixo) para grafos de diferentes tamanhos, dado por n . Em particular, considere $n = \{10, 50, 100, 300, 700, 1000\}$.

4.1 Trace um único gráfico de τ_n em função de n onde cada curva corresponde a um grafo.

O gráfico da questão é apresentado na Figura 10.

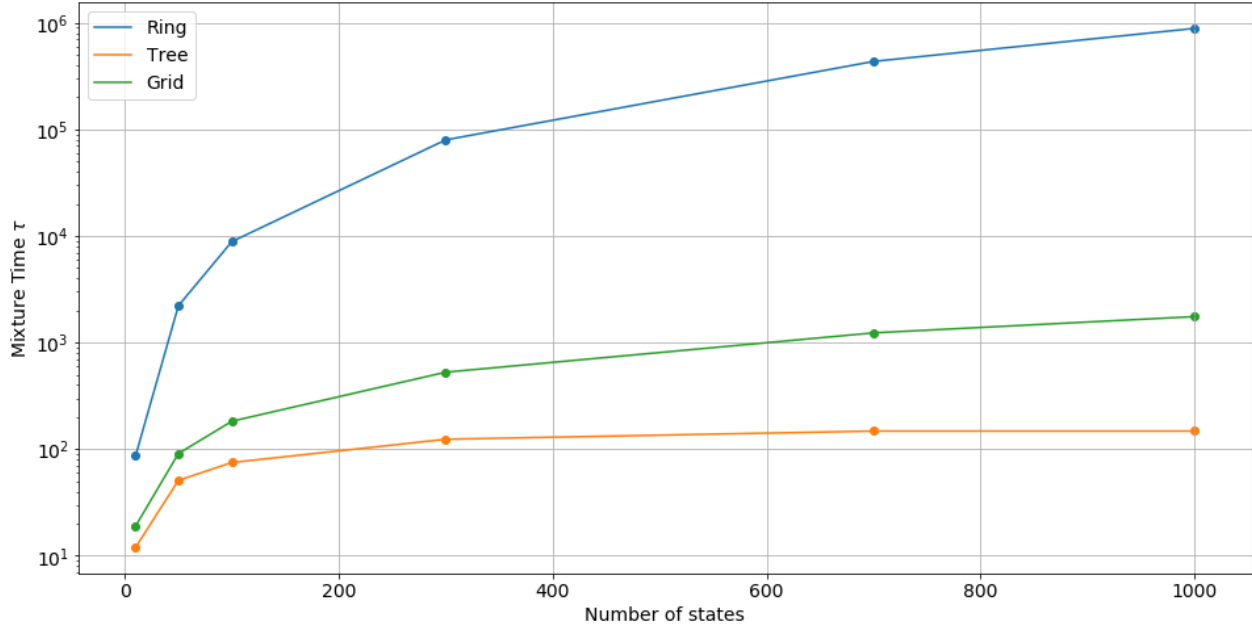


Figure 10: Tempo de mistura para diferentes quantidades de estados nos grafos em anel, árvore binária e reticulado.

4.2 O que você pode concluir sobre o tempo de mistura em função da estrutura do grafo?

Podemos perceber que o tempo de mistura é maior nos grafos em anel, reticulado e árvore binária, nesta ordem, sendo consideravelmente maior para o anel.

Como citado anteriormente, os grafos em anel e reticulado possuem a mesma distribuição estacionária. No entanto, a distribuição de graus é diferente para cada um deles: enquanto no anel todos os vértices têm grau 3, no reticulado todos os vértices possuem grau 5. Isso faz com que o passeio possa chegar em mais vértices com menos iterações e, consequentemente, para cada estado, a fração de vezes que o estado é visitado convirja mais rápido.

Para o grafo em árvore binária, por outro lado, podemos destacar o efeito da centralidade do nó raiz na convergência. Note que, separando-se o grafo pela metade, para passar de uma metade do grafo para outra, necessariamente o passeio precisa passar pelo nó-raiz. Isso faz com que qualquer caminho entre vértices de metades diferentes precise passar pelo nó raiz. Assim, o passeio fica concentrado em uma primeira metade da árvore e, uma vez que ele passa para a outra metade, ele também fica concentrado até que o passeio atinja o nó-raiz, quando o processo se repete.

5 Hardcore com Metropolis-Hasting: Considere o modelo Hardcore, onde cada vértice de um grafo qualquer está associado ao valor 0 ou 1 tal que um vértice com valor 1 necessariamente tem todos seus vizinhos com valor 0.

5.1 Construa uma cadeia de Markov irreduzível e aperiódica que tenha transições simples de calcular entre os possíveis estados do modelo. Por exemplo, o número de transições de saída de um estado deve ser no máximo o número de vértices da rede.

Considere S o conjunto de configurações possíveis do modelo Hardcore e S^+ o subconjunto de S com as configurações válidas, isto é, quando não há vértices vizinhos com valores 1. Desta forma, os elementos de S^+ são os estados da cadeia de Markov que estamos interessados. Para um grafo com N vértices, temos um total de 2^N configurações possíveis, pois cada vértice pode ter seu valor igual a 0 ou 1. Assim, $|S| = 2^N$.

Queremos realizar amostras de configurações válidas no espaço S^+ . Para isso, dado que estamos em um estado válido X_i , queremos transitar para a próxima configuração válida X_j a partir da probabilidade de transição p_{ij} entre as mesmas.

Podemos fazer isso amostrando-se uniformemente um vértice v da configuração X_i e verificando se todos os seus vizinhos possuem valor 0. Caso esta condição seja satisfeita, então associa-se a v o valor contrário ao seu valor atual com probabilidade p e com probabilidade $1 - p$ não se altera o valor. Desta forma, se $v = 1$ em X_i , então $v = 0$ com probabilidade p (e portanto, passamos para o estado X_j) ou $v = 1$ com probabilidade $1 - p$ (e portanto não mudamos de estado).

Portanto, dado i e j configurações válidas que se diferenciam de 0 ou 1 vértices, então p_{ij} é dado por:

$$p_{ij} = \begin{cases} \frac{1}{N}p & \text{se } i \neq j \\ \frac{1}{N}(1 - p) & \text{se } i = j \end{cases}$$

Para outras configurações onde i e j se diferenciam em mais de 1 vértice, considere $p_{ij} = 0$. Portanto, por esta configuração, temos que a cadeia é simétrica, pois $p_{ij} = p_{ji}$.

Note que, ao escolher uniformemente um vértice aleatório e mudando-se o seu valor de forma que o próximo estado seja válido, o número de estados possíveis que se possa ir é N . Por exemplo, para o caso da configuração onde todos os vértices estão em 0, qualquer um dos N vértices que sejam escolhidos satisfaz a condição de ter todos os seus vizinhos em branco e, portanto, pode levar a uma nova configuração válida.

Ainda, note que os estados possuem *self-loops*, o que garante que a cadeia seja aperiódica.

5.2 Aplique o método de Metropolis-Hasting na CM acima tal que a distribuição estacionária da nova cadeia seja uniforme. Calcule as probabilidades de transição desta nova cadeia.

Queremos induzir uma cadeia de forma que os estados tenham probabilidade uniforme de serem amostrados. Assim,

$$\pi_i = \frac{1}{|S^+|}$$

Onde $|S^+|$ é o número de possíveis configurações válidas, que é constante. Portanto, aplicando-se o algoritmo de Metropolis-Hasting, temos:

$$a_{ij} = \min\left\{1, \frac{\pi_i p_{ij}}{\pi_j p_{ji}}\right\} = \min\{1, 1\} = 1$$

Note que a probabilidade de aceitar a transição é 1, o que faz com que a cadeia de Markov final seja igual à original:

$$p'_{ij} = p_{ij} \cdot a_{ij} = p_{ij}$$

Fazendo com que $P' = P$.

6 Amostrando triângulos: Considere um grafo conexo qualquer. Desejamos gerar amostras de triângulos deste grafo (cliques de tamanho 3), tal que todo triângulo tem igual probabilidade de ser amostrado. Ou seja, distribuição uniforme sobre o conjunto de triângulos do grafo.

6.1 Mostre como gerar amostras de forma direta, utilizando a distribuição uniforme (dica: pense em amostragem por rejeição). Determine a eficiência desse método.

Considere S o conjunto de todas as triplas de vértices do grafo. Ou seja, dado um grafo com N vértices,

$$|S| = \binom{n}{3}$$

Assim, o conjunto de interesse S^+ é um subconjunto de S . As amostras de S pode ser feitas amostrando-se uniformemente 3 vértices do grafo. Logo, considerando um elemento $s \in S$, a probabilidade dele ser amostrado é:

$$P[s = i] = \frac{1}{\binom{n}{3}} = \frac{1}{|S^+|} = \pi_i$$

Seja $I(s)$ uma variável aleatória indicadora tal que:

$$I(s) = \begin{cases} 1 & \text{se } s \text{ é um triângulo} \\ 0 & \text{se } s \text{ é um quase-triângulo} \end{cases}$$

Assim, podemos ter uma sequência de amostras rejeitadas até que um triângulo apareça. Esta formulação faz com que a amostragem dos triângulos se assemelhe à amostragem de uma geométrica, uma vez que estamos amostrando triplas de vértices até que um triângulo apareça. Em particular, o parâmetro p da geométrica é igual à fração de triângulos em S .

Uma característica desse método é que ele irá rejeitar muitas triplas até aparecer um triângulo. Assim, dependendo da rede, o método pode ser ineficiente.

Por outro lado, podemos melhorar a eficiência deste método modificando a amostragem dos vértices em S . Sabemos que em um grafo a clusterização de um vértice está relacionada à probabilidade de seus vizinhos também serem vizinhos. Para redes reais, por exemplo, essas clusterizações tendem a ser baixíssimas - ou seja, poucos triângulos ou quase-triângulos são formados.

Assim, uma forma de melhorar a eficiência do algoritmo é amostrar os vértices do grafo proporcionalmente à sua clusterização. Desta forma, intuitivamente a chance de haver um triângulo aumenta e a taxa de rejeição diminui.

6.2 Mostre como gerar amostras utilizando Metropolis-Hasting. Determine os estados da CM, as transições originais da cadeia (que deve ser irredutível e aperiódica), e as transições da cadeia modificada pelo método Metropolis-Hasting.

Considere a cadeia onde cada estado é uma configuração de tripla de vértices de todas as combinações possíveis. Assim, dado que estamos em um estado i , podemos definir os estados vizinhos j como os estados que diferem de 1 vértice do estado i .

Assim, seja o estado i composto pelos vértices $\{v_1, v_2, v_3\}$. Para transitarmos para o estado j , podemos escolher um vértice uniformemente em i e substituí-lo por outro vértice amostrado uniformemente do conjunto de vértices do grafo, com exceção dos vértices já inseridos no estado i . Assim, o vértice a ser substituído é escolhido com probabilidade $1/3$ e o vértice a ser inserido é escolhido com probabilidade $1/(n-2)$, o que faz com que a probabilidade de transição do estado i para o j seja:

$$p_{ij} = \frac{1}{3} \frac{1}{n-2}$$

6.3 Intuitivamente, discuta quando a abordagem via Metropolis-Hastings é mais eficiente (do ponto de vista computacional) do que a abordagem via amostragem por rejeição.

Depende. Intuitivamente, quando que fazer uma amostragem por rejeição pode ser mais eficiente do que fazer uma cadeia de markov, esperar o tempo de mistura, etc.?

Se tivermos uma rede com muitos triangulos, muito provavelmente conseguiremos selecionar de maneira uniforme 3 vértices ao acaso. Em contrapartida, se tivermos poucos triangulos na rede (grau de clusterização baixo), então vale mais a pena gerar uma cadeia de markov e percorrê-la.

7 Quebrando o código: Você encontrou uma mensagem que foi cifrada com o código da substituição (neste código, cada letra é mapeada em outra letra, de forma bijetora). Você deseja encontrar a chave do código para ler a mensagem. Repare que a chave é um mapeamento σ entre as letras, por exemplo $\sigma(a) = x, \sigma(b) = h, \sigma(c) = e, \dots$. Considere uma função $f : \Omega \rightarrow [0, 1]$ que avalia a sua capacidade de entender a mensagem cifrada dado um mapeamento $\sigma \in \Omega$. Repare que $f(\sigma) = 1$ se você consegue entender por completo a mensagem decifrada com o mapeamento σ , e $f(\sigma) = 0$ se o mapeamento σ não revela nenhuma informação sobre a mensagem. Utilize o método de Simulated Annealing para resolver este problema! Mostre todos os passos necessários.

Seja um vetor de 26 posições onde serão alocados caracteres no vetor. O valor da posição 0 é mapeado na letra “a”, o da posição 1 na letra “b”, e assim por diante em ordem alfabética até que o valor na posição 25 seja mapeado na letra “z”. Uma ilustração pode ser vista na Figura 11.

X	H	E	L
A	B	C	Z

Figure 11: Representação do vetor de mapeamento.

Assim, podemos considerar uma cadeia de Markov onde cada estado seja uma permutação deste vetor de mapeamento. Uma vez que o mapeamento pode ser avaliado através da função $f(\sigma)$, então podemos mensurar o valor de um estado.

A transição entre os estados pode ser feita trocando-se o mapeamento de 2 posições distintas. Por exemplo, dado o mapeamento $[\sigma(a) = x, \sigma(b) = h]$, um novo estado é obtido por $[\sigma(a) = h, \sigma(b) = x]$. Logo, para escolher as posições i e j de inversão do mapeamento uniformemente, temos:

$$P[(i, j)] = \frac{1}{\binom{n}{2}} = \frac{1}{\binom{26}{2}} = \frac{2}{26(26-1)}$$

onde $\binom{26}{2}$ é o número de pares de posições possíveis.

Assim, podemos andar numa cadeia onde os vizinhos de um estado diferem dele em 2 posições de mapeamento, e com isso, a probabilidade de ir do estado s para s' é dada por:

$$P_{ss'} = \frac{2}{26(26-1)}$$

No entanto, podemos utilizar o algoritmo de Metropolis-Hasting para modificar as probabilidades de transição de forma a ir para o estado de maior valor. Para isso, consideramos uma cadeia de Markov modificada de forma que:

$$\pi_s = \frac{e^{+f(s)/T}}{Z}$$

onde $Z = \sum_{s \in S} e^{+f(s)/T}$ é uma constante de normalização que considera o conjunto de todos os estados possíveis S . Assim, as probabilidades de transições na nova cadeia de Markov são dadas por:

$$P'_{ss'} = P_{ss'} \min\left\{1, \frac{\pi_{s'}}{\pi_s}\right\} = \frac{2}{n(n-1)} \min\left\{1, e^{\frac{f(s')-f(s)}{T}}\right\}$$

Note que não precisamos calcular o valor da constante de normalização Z , o que é uma das grandes vantagens da aplicação do Metropolis-Hasting, onde conseguimos gerar amostras do espaço de estado sem conhecer

todos os estados.

Como podemos ver, este modelo segue um parâmetro T chamado de *temperatura de resfriamento*. Para cada valor de T , teremos uma cadeia de Markov com os mesmos estados, mas com diferentes probabilidades de transição P' . Em particular, para $T \rightarrow 0$, teremos uma importância ainda maior da diferença entre $f(s)$ e $f(s')$ no cálculo das probabilidades.

Podemos usar o controle do parâmetro T para impedir que o passeio siga um caminho que leve a máximos locais na busca pelo mapeamento ótimo. Isto pode ser feito modificando-se a temperatura T a cada N passos na cadeia de Markov induzida. Diferentes estratégias podem ser usadas aqui para escolher como a temperatura varia e quantos passos serão dados em cada temperatura.