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

4ª Lista de Exercícios

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

29 de maio de 2025

Questão 1: Sequências binárias restritas

Considere uma sequência de dígitos binários (0s e 1s) de comprimento s. Uma sequência é dita válida se ela não possui 1s adjacentes. Considerando a distribuição uniforme, queremos determinar o valor esperado do número de 1s de uma sequência válida, denotado por μ_s.

- Considerando $s = 4$, determine todas as sequências válidas e calcule μ_4.

As sequências válidas de comprimento 4 são: 0000, 0001, 0010, 0100, 1000, 0101, 1010, 1001. Portanto, temos um total de 8 sequências válidas. O valor esperado de 1s nessas sequências é dado por:

$$\mu_4 = \frac{1}{8} \times 0 + \frac{4}{8} \times 1 + \frac{3}{8} \times 2 = \frac{10}{8}$$

Assim,

$$\boxed{\mu_4 = \frac{10}{8} \approx 1.25}$$

- Construa uma cadeia de Markov sobre o conjunto de sequências válidas, deixando claro como funcionam as transições de estado. Argumente que a cadeia é irredutível.

A cadeia de Markov pode ser construída considerando os estados como as sequências válidas (ou seja, aquelas de comprimento $s = 4$ que não possuem dígitos 1 adjacentes). As transições entre estados ocorrem ao flipar um único dígito da sequência, desde que o resultado continue sendo uma sequência válida.

Por exemplo, a sequência 0010 pode transitar para 0000 (flipando o terceiro dígito) ou para 1010 (flipando o primeiro dígito), pois ambas são válidas. Por outro lado, flipar o segundo dígito resultaria em 0110, que não é válida (pois possui 1s adjacentes), portanto essa transição não é permitida.

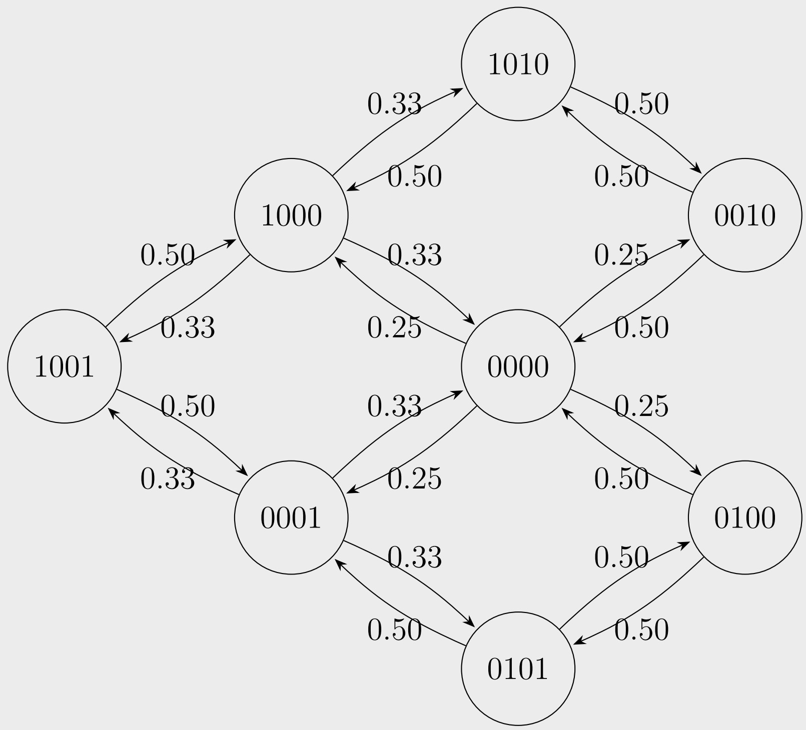
A probabilidade de transição do estado i para j é dada por:

$$P_{ij} = \begin{cases} \frac{1}{d_i} & \text{se } j \text{ é obtido de } i \text{ por flip válido de um dígito} \\ 0 & \text{caso contrário} \end{cases}$$

onde d_i é o número total de transições válidas a partir do estado i.

A cadeia é irredutível porque, para qualquer par de estados válidos, é possível transformar um no outro através de uma sequência finita de flips de dígitos (desde que se respeite a restrição de não criar 1s adjacentes). Como todas as sequências válidas estão conectadas por essas transições, a cadeia é conexa e, portanto, irredutível.

- Desenhe a cadeia de Markov para o caso de $s = 4$, mostrando todas as transições.



- Mostre como aplicar Metropolis-Hastings para resolver o problema de estimar μ_s. Deixe claro as probabilidades de aceite e o funcionamento do estimador.

Partindo da cadeia de Markov construída anteriormente, podemos aplicar o algoritmo de Metropolis-Hastings para modificar suas probabilidades de transição, de modo que a nova cadeia tenha como distribuição estacionária a distribuição uniforme sobre o conjunto de sequências válidas. A nova matriz de probabilidade de transição P' é dada por:

$$P'_{ij} = \begin{cases} P_{ij}a(i,j) & \text{se } j \text{ é obtido de } i \text{ por flip válido de um dígito} \\ 1 - \sum_{k:k\neq i} P_{ik}a(i,k) & \text{se } i = j \\ 0 & \text{caso contrário} \end{cases}$$

onde $a(i,j)$ é a probabilidade de aceitar a transição de i para j. Essa probabilidade pode ser encontrada pela equação da condição de reversibilidade:

$$\pi_i P_{ij} a(i,j) = \pi_j P_{ji} a(j,i)$$

Como temos uma equação é duas incógnitas($a(i,j)$ e $a(j,i)$), precisamos arbitrar um valor para uma delas. Arbitrando $a(i,j) = 1$ se $\pi_i P_{ij} \leq \pi_j P_{ji}$ e, consequentemente, $a(i,j) = \frac{\pi_j P_{ji}}{\pi_i P_{ij}}$ se $\pi_i P_{ij} > \pi_j P_{ji}$, temos que:

$$a(i,j) = \min\{1, \frac{\pi_j P_{ji}}{\pi_i P_{ij}}\}$$

Como a distribuição estacionária é uniforme ($\pi_i = \pi_j$) podemos simplificar a equação para:

$$a(i,j) = \min\{1, \frac{P_{ji}}{P_{ij}}\} = \min\{1, \frac{\frac{1}{d_j}}{\frac{1}{d_i}}\} = \min\{1, \frac{d_i}{d_j}\}$$

A nova matriz de probabilidade de transição P' fica:

$$P'_{ij} = \begin{cases} \frac{1}{d_i} \min\{1, \frac{d_i}{d_j}\} & \text{se } j \text{ é obtido de } i \text{ por flip válido de um dígito} \\ 1 - \sum_{k:k \neq i} P_{ik} a(i,k) & \text{se } i = j \\ 0 & \text{caso contrário} \end{cases}$$

Assim, podemos amostrar de forma uniforme as sequências válidas seguindo o seguinte procedimento:

1. Descobrir os d_i vizinhos do estado atual i;
2. Escolher de forma uniforme um vizinho j e contar os seus d_j vizinhos;
3. Gerar um número aleatório u entre 0 e 1;
4. Aceitar a transição se $u < a(i,j) = \min\{1, \frac{d_i}{d_j}\}$. Caso contrário, repetir estado i;
5. Repetir o processo a partir do novo estado.

Após gerar um número suficiente de amostras, o teorema de ergodicidade garante que a média das amostras converge para o valor esperado μ_s. Assim, podemos estimar μ_s como:

$$\hat{\mu}_s = \frac{1}{N} \sum_{i=1}^{N} X_i$$

onde X_i é o número de 1s na amostra i e N é o número total de amostras geradas.

Implementando o procedimento acima em Python (link para o código no final do relatório), obtemos os seguintes resultados para $N = 10^5$ e diferentes valores de s:

s	n° estados	$\hat{\mu}_s$
4	8	1.2479
8	55	2.3695
12	377	3.4821
16	2584	4.5997

Questão 2: Amostras de Modelos de Mistura

Considere a seguinte função de probabilidade:

$$p(x) = \alpha p_B(x; n, p_1) + (1 - \alpha)p_B(x; n, p_2),$$

onde $p_B(x; n, p)$ é a probabilidade associada ao valor x da binomial com parâmetros n e p, e $\alpha \in [0, 1]$ é um peso. Trata-se de um modelo de mistura de duas binomiais com diferentes valores de p, com pesos dados por α e $1 - \alpha$. Considere duas variáveis aleatórias X e K, representando o valor de $X \in [0, n]$ e a binomial utilizada $K \in \{1, 2\}$. Queremos gerar amostras de acordo com $p(x)$.

- Determine as distribuições de probabilidade condicionais $P(X|K)$ e $P(K|X)$. Dica: utilize a regra de Bayes no segundo caso.

> A distribuição de X dado K é simplesmente uma binomial com parâmetro p_k, para $k \in \{1, 2\}$:
>
> $$P(X = x|K = k) = p_B(x; n, p_k)$$
>
> $$\boxed{P(X = x|K = k) = \binom{n}{x} p_k^x (1 - p_k)^{n-x}}$$
>
> A distribuição de K dado X é obtida pela regra de Bayes:
>
> $$P(K = k|X = x) = \frac{P(X = x|K = k) \cdot P(K = k)}{p(x)}$$
>
> Como $P(K = 1) = \alpha$ e $P(K = 2) = 1 - \alpha$, temos:
>
> $$\boxed{P(K = k|X = x) = \frac{p_B(x; n, p_k) \cdot p(K = k)}{\alpha p_B(x; n, p_1) + (1 - \alpha)p_B(x; n, p_2)}}$$

- Determine a distribuição de probabilidade conjunta $P(X, K)$.

> A distribuição conjunta é dada pela regra do produto:
>
> $$P(X = x, K = k) = P(X = x|K = k) \cdot P(K = k)$$
>
> Como $P(X = x|K = k) = p_B(x; n, p_k)$, $P(K = 1) = \alpha$, $P(K = 2) = 1 - \alpha$, temos:
>
> $$\boxed{P(X = x, K = k) = p_B(x; n, p_k) \cdot P(K = k)}$$

- Utilize a técnica de Gibbs Sampling para gerar amostras de X. Mostre como construir a cadeia de Markov e determine a transição entre os estados.

  Cada estado da cadeia de Markov é representado por um par $V = (X, K)$, onde $X \in [0, n]$ é o valor da variável aleatória e $K \in \{1, 2\}$ indica qual das duas binomiais foi utilizada.

  As transições entre os estados ocorrem por meio da atualização de uma única variável por vez, mantendo a outra fixa:

  – Para transições em que apenas X é atualizado (com K fixo), a probabilidade de transição para o novo estado (X', K) é dada por:

  $$P_{(X,K),(X',K)} = \frac{1}{2} P[X'|K] = \frac{1}{2} \cdot p_B(X'; n, p_K)$$

  – Para transições em que apenas K é atualizado (com X fixo), a probabilidade de transição para o novo estado (X, K') é dada por:

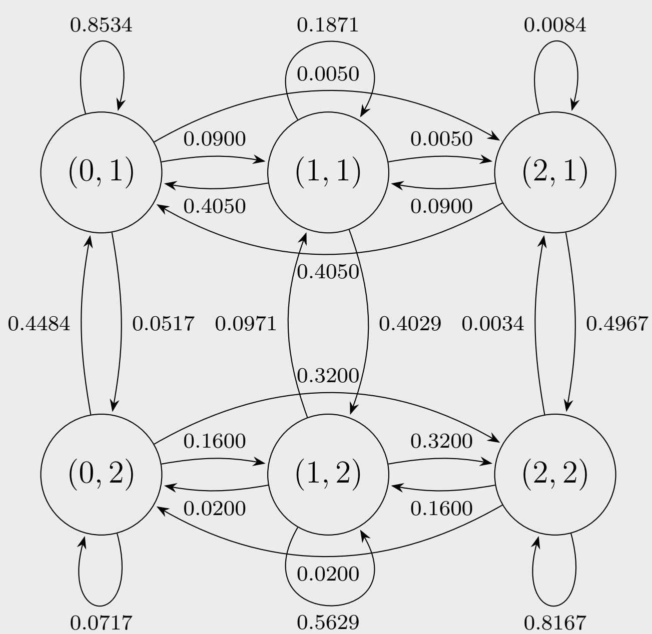
  $$P_{(X,K),(X,K')} = \frac{1}{2} P[K'|X] = \frac{1}{2} \cdot \frac{p_B(X; n, p_{K'}) \cdot P(K')}{\alpha \cdot p_B(X; n, p_1) + (1 - \alpha) \cdot p_B(X; n, p_2)}$$

  onde $P(K' = 1) = \alpha$ e $P(K' = 2) = 1 - \alpha$.

  Dessa forma, a cadeia se move no espaço de pares (X, K), com transições definidas pelas distribuições condicionais do modelo de mistura. A estrutura garante que a cadeia seja reversível e tenha como distribuição estacionária a distribuição conjunta $P(X, K)$.

- Para $n = 2$, $p_1 = 0{,}1$, $p_2 = 0{,}8$, $\alpha = 0{,}3$, desenhe a cadeia de Markov com todas as transições.

Probabilidade	Valor	Valor/2
$P(X = 0 \mid K = 1)$	0.8100	0.4050
$P(X = 1 \mid K = 1)$	0.1800	0.0900
$P(X = 2 \mid K = 1)$	0.0100	0.0050
$P(X = 0 \mid K = 2)$	0.0400	0.0200
$P(X = 1 \mid K = 2)$	0.3200	0.1600
$P(X = 2 \mid K = 2)$	0.6400	0.3200
$P(K = 1 \mid X = 0)$	0.8967	0.4484
$P(K = 2 \mid X = 0)$	0.1033	0.0517
$P(K = 1 \mid X = 1)$	0.1942	0.0971
$P(K = 2 \mid X = 1)$	0.8058	0.4029
$P(K = 1 \mid X = 2)$	0.0067	0.0034
$P(K = 2 \mid X = 2)$	0.9933	0.4967



OBS: os valores das probabilidades dos selfloops são a soma das probabilidades de transição mantendo X fixo e mantendo K fixo, ou seja, $P(X = x|K = k) + P(K = k|X = x)$.

- Descreva como utilizar a cadeia de Markov para gerar amostras.

1. Inicializar a cadeia em um estado arbitrário (X, K), por exemplo $(0, 1)$.
2. Escolher aleatoriamente qual variável será atualizada:
   - Com probabilidade $\frac{1}{2}$, atualizar X;
   - Com probabilidade $\frac{1}{2}$, atualizar K.
3. Atualizar a variável escolhida:
   - Se for X, sorteie X' da distribuição binomial $P(X \mid K)$ com parâmetro p_K;
   - Se for K, sorteie K' da distribuição $P(K \mid X)$, calculada pela regra de Bayes.

   A variável não escolhida permanece inalterada.
4. Atualizar o estado da cadeia para o novo par (X', K').
5. Repitir o processo. Após τ_ϵ passos (tempo de mistura), o valor de X pode ser considerado uma amostra da distribuição desejada.
6. Repitir os passos acima para obter quantas amostras forem necessárias, aguardando sempre pelo menos τ_ϵ passos entre duas amostras consecutivas.

O tempo de mistura τ_ϵ pode ser limitado superiormente por:

$$\tau_\epsilon \leq \frac{\log 1/(\pi_o \epsilon)}{\delta}$$

onde π_o é a probabilidade do estado menos provável na distribuição estacionária da cadeia de Markov, ϵ é a tolerância desejada e δ é o vão espectral, definido por:

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

onde λ_2 é o segundo maior autovalor da matriz de transição da cadeia de Markov.

Questão 3: Amostrando triângulos

Considere um grafo conexo qualquer. Desejamos gerar amostras de triângulos deste grafo (cliques de tamanho 3), tal que todo triângulo tenha igual probabilidade de ser amostrado – ou seja, uma distribuição uniforme sobre o conjunto de triângulos do grafo.

- Mostre como gerar amostras de forma direta, utilizando a distribuição uniforme. Dica: pense em amostragem por rejeição. Determine a eficiência desse método.

O procedimento de amostragem por rejeição consiste em gerar amostras de um espaço amostral maior e, em seguida, rejeitar aquelas que não atendem a um critério específico. Neste caso, o espaço amostral maior é o conjunto de todas as combinações possíveis de 3 vértices do grafo. Para cada combinação, verificamos se os 3 vértices formam um triângulo (ou seja, se estão todos conectados entre si). Se formarem um triângulo, aceitamos a amostra; caso contrário, rejeitamos.

O procedimento pode ser descrito da seguinte forma:

1. Escolher três vértices distintos $u, v, w \in V$ uniformemente ao acaso.
2. Verificar se o conjunto $\{u, v, w\}$ forma um triângulo, ou seja, se as três arestas (u, v), (v, w) e (u, w) pertencem ao conjunto de arestas E.
3. Se os vértices formam um triângulo, aceitar a amostra e registrar o triângulo $\{u, v, w\}$.
4. Caso contrário, rejeitar e voltar ao passo 1.
5. Repetir o processo até obter o número desejado de amostras.

A eficiência do método é dada por:

$$e = \frac{T}{\binom{|V|}{3}}$$

onde T é o número de triângulos no grafo, $|V|$ é a quantidade de vértices e $\binom{|V|}{3}$ representa o número total de trios possíveis de vértices. Assim, a eficiência do método depende da densidade do grafo e do número de triângulos presentes: se o grafo for esparso, a eficiência será baixa; se o grafo for denso, a eficiência será maior.

- Mostre como gerar amostras utilizando Metropolis-Hastings. Determine os estados da cadeia de Markov, as transições da cadeia base (que deve ser irredutível) e a probabilidade de aceitação na cadeia modificada pelo método Metropolis-Hastings.

Podemos criar uma Cadeia de Markov na qual os estados são caminhos de tamanho 3, ou seja, conjuntos de 3 vértices $\{u, v, w\}$, onde, pelo menos uv e vw são arestas do grafo.

A regra de transição da cadeia base pode ser definida como a remoção de um dos vértices da ponta (u ou w) e a adição de um novo vértice que esteja conectado ao vértice da outra ponta. Assim, um estado $\{u, v, w\}$ pode transicionar para $\{v, w, x\}$ se escolhermos remover u e escolher x tal que é vizinho de w, ou pode transicionar para $\{x, u, v\}$, se escolhermos remover w e escolher x tal que é vizinho de u. A cadeia é irredutível porque, a partir de qualquer caminho de comprimento 3, podemos alcançar qualquer outro caminho de comprimento 3 removendo um vértice de uma ponta e adicionando outro conectado na outra ponta.

A probabilidade de transição entre dois estados é dada por:

$$P_{s \to s'} = \frac{1}{d-1}$$

onde d é o grau do vértice da ponta contrária a que foi removida (u ou w) do estado atual menos 1 (pois não podemos escolher o vértice v que já está no meio do caminho de comprimento 2 e irá para uma das pontas no próximo estado).

A probabilidade de transição na cadeia modificada pelo método Metropolis-Hastings é dada por:

$$P'_{s \to s'} = \frac{1}{d-1} \min\left(1, \frac{d-1}{d'}\right)$$

onde d' é o grau do novo vértice que será adicionado ao caminho no estado s'.

É fácil ver que nem todos os estados da Cadeia de Markov são necessariamente triângulos, mas todos os triângulos são alcançados a partir de algum estado da cadeia. Então podemos condicionar o processo de amostragem para que apenas triângulos sejam aceitos. Como a Cadeia de Markov modificada possui distribuição estacionária uniforme, a distribuição dessa amostragem condicional também será uniforme.

O procedimento de amostragem pode ser descrito da seguinte forma:

1. Escolher um caminho de comprimento 2 inicial $s = \{u, v, w\}$ uniformemente ao acaso.
2. Executar o procedimento abaixo:
   (a) Escolher com probabilidade $\frac{1}{2}$ remover o primero vértice (u) ou o terceiro vértice (w).
   (b) Escolher com probabilidade uniforme $\frac{1}{d-1}$ um novo vértice x que seja vizinho do vértice da outra ponta (ou seja, u se w foi removido ou w se u foi removido).
   (c) Gerar um número aleatório r uniformemente no intervalo $[0, 1]$.
   (d) Se $r < \min\left(1, \frac{d-1}{d'}\right)$, aceitar a amostra e registrar s', senão, repetir o estado anterior fazendo $s' = s$.
3. Repetir os passos $a - d$ por τ_ϵ iterações, onde τ_ϵ é o tempo de mistura da Cadeia de Markov.

4. Verificar se o último estado s' é um triângulo, ou seja, se as arestas (u', v'), (v', w') e (u', w') pertencem ao conjunto de arestas E. Caso sim, aceitar a amostra e registrar o triângulo s', caso o contrário, continuar repetindo os passos $a - d$ até obter um triângulo.
5. Repetir os passos $2 - 4$ até obter o número desejado de amostras de triângulos.

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

A abordagem via Metropolis-Hastings tende a ser mais eficiente que a amostragem por rejeição quando o grafo é esparso, ou seja, quando o número de triângulos T é pequeno em relação ao número total de combinações de 3 vértices $\binom{|V|}{3}$. Nesse cenário, a amostragem por rejeição desperdiça muitas tentativas, pois a chance de um trio aleatório de vértices formar um triângulo é muito baixa. Como resultado, o número de rejeições cresce rapidamente, tornando o método ineficiente em tempo de execução.

Por outro lado, a abordagem via Metropolis-Hastings opera sobre caminhos de comprimento 2, ou seja, trios de vértices onde duas das três conexões já existem por construção. Isso significa que, em cada iteração, a cadeia considera apenas trios parcialmente conectados, nos quais falta apenas uma aresta para formar um triângulo. Assim, a chance de o trio atual ser um triângulo é muito maior do que na amostragem por rejeição, que escolhe vértices completamente ao acaso. Dessa forma, mesmo rejeitando trios que não formam triângulos, o processo explora mais eficientemente o espaço de triângulos do grafo.

Questão 4: Quebrando o código

Você encontrou uma mensagem cifrada com o código de substituição (neste código, cada letra é mapeada em outra letra de forma bijetiva). Você deseja encontrar a chave do código para ler a mensagem. Repare que a chave é um mapeamento σ entre as letras, por exemplo $\sigma(a) = x$, $\sigma(b) = h$, $\sigma(c) = e$, etc. Considere uma função $f : \Omega \to [0, 1]$ que avalia a capacidade de uma pessoa entender a mensagem cifrada dado um mapeamento $\sigma \in \Omega$. Repare que $f(\sigma) = 1$ significa que é possível entender por completo a mensagem decifrada com o mapeamento σ, e $f(\sigma) = 0$ se o mapeamento não revela nenhuma informação sobre a mensagem. Mostre como a técnica de Simulated Annealing pode ser utilizada para ler a mensagem cifrada. Mostre todos os passos necessários para aplicar a técnica neste problema (não é necessário implementar).

O mapeamento σ pode ser representado como uma permutação das letras do alfabeto. No exemplo do enunciado, se temos $\sigma(a) = x$, $\sigma(b) = h$, $\sigma(c) = e$. Se o alfabeto é $\{a, b, c, \ldots\}$, podemos representar $\sigma = \{x, h, e, \ldots\}$.

Seja $f : \Omega \to [0, 1]$ a função que avalia a capacidade de entender a mensagem cifrada com o mapeamento σ. Esta função pode ser implementada como a saída de um modelo de linguagem, que atribui uma pontuação à legibilidade da mensagem decifrada. Por exemplo, se a mensagem decifrada com o mapeamento σ é gramaticalmente correta e faz sentido, $f(\sigma)$ será próximo de 1; caso contrário, será próximo de 0.

O algoritmo de Simulated Annealing pode ser aplicado criando-se uma Cadeia de Markov base na qual cada estado será uma permutação do alfabeto, ou seja, um mapeamento σ que associa cada letra a outra letra. Considerando o alfabeto com 26 letras, o espaço de estados Ω terá 26! permutações possíveis.

As transições entre os estados da cadeia base podem ser feitas invertendo partes da permutação, escolhendo índices aleatórios i e j em $\{1, 2, \ldots, 26\}$, com $i < j$ e invertendo a ordem das letras entre esses índices. A inversão de letras entre i e j permite alterar várias posições da permutação ao mesmo tempo, o que ajuda a escapar de máximos locais. A probabilidade de transição independe da permutação atual, pois é uma escolha sem repetição entre $\{1, 2, \ldots, 26\}$, ou seja, $n(n-1)/2 = 325$ possibilidades uniformes de transição, com $n = 26$. Como todos os estados têm o mesmo grau de saída e as mesmas probabilidades de transição, a cadeia é simétrica.

A distribuição estacionária da cadeia deve ser definida pela distribuição de Boltzmann, que é dada por:

$$\pi_\sigma = \frac{e^{\frac{f(\sigma)}{T}}}{Z}$$

onde T é um parâmetro, chamado de temperatura, e $Z = \sum_{\sigma \in \Omega} e^{\frac{f(\sigma)}{T}}$ é a constante de normalização, que garante que a soma das probabilidades seja 1.

Utilizando agora o algoritmo de Metropolis-Hastings para modificar a cadeia base, podemos definir a probabilidade de transição entre dois estados σ e σ' como:

$$P'(\sigma, \sigma') = P(\sigma, \sigma') \min \left(1, \frac{\pi'_\sigma P_(\sigma', \sigma)}{\pi_\sigma P_(\sigma, \sigma')}\right) = \frac{1}{325} \min \left(1, \frac{\frac{e^{\frac{f(\sigma')}{T}}}{Z} \frac{1}{325}}{\frac{e^{\frac{f(\sigma)}{T}}}{Z} \frac{1}{325}}\right) = \frac{1}{325} \min \left(1, e^{\frac{f(\sigma') - f(\sigma)}{T}}\right)$$

onde o termo $\frac{1}{325}$ é a probabilidade de transição uniforme entre os estados, e o termo $\min \left(1, e^{\frac{f(\sigma') - f(\sigma)}{T}}\right)$ é a probabilidade de aceitação da transição.

Para definir a agenda de resfriamento (*annealing*), devemos testar diferentes estratégias e escolher a que melhor se adapta ao problema. Uma boa estratégia para começar é definir apenas um passo para cada temperatura ($N_t = 1$) e reduzir a temperatura de forma logarítmica, com $T_t = a/\log(t + b)$, onde a e b são parâmetros a serem ajustados, pois esta estratégia possui uma prova de convergência global se a for alta o suficiente e b for constante.

O algoritmo de Simulated Annealing para decifrar a mensagem cifrada pode ser descrito da seguinte forma:

1. Inicializar a temperatura T_0, escolher uma permutação inicial σ_0 aleatória, fazer $\sigma_* = \sigma_0$.
2. Calcular $f(\sigma_0)$.
3. Para cada iteração t:
   (a) Escolher i e j aleatórios em $\{1, 2, \ldots, 26\}$ com $i < j$.
   (b) Gerar uma nova permutação σ' invertendo as letras entre os índices i e j em σ_t.
   (c) Calcular $f(\sigma')$.
   (d) Se $f(\sigma') > f(\sigma_t)$, aceitar a transição com probabilidade 1. Caso o contrário, aceitar a transição com probabilidade $e^{\frac{f(\sigma') - f(\sigma_t)}{T_t}}$.
   (e) Se aceitar, atualizar $\sigma_{t+1} = \sigma'$; caso contrário, manter $\sigma_{t+1} = \sigma_t$.
   (f) Se $f(\sigma_{t+1}) > f(\sigma_*)$, atualizar $\sigma_* = \sigma_{t+1}$.
   (g) Atualizar a temperatura $T_{t+1} = a/\log(t + b)$.
4. Repetir $(a) - (g)$ até que a temperatura atinja um valor mínimo ou até que um outro critério de parada seja atingido (por exemplo, um número máximo de iterações ou um valor de $f(\sigma_*)$ próximo o suficiente de 1).
5. Retornar σ_* como a melhor permutação encontrada, que melhor decifra a mensagem cifrada.

Códigos

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