Algoritmos

Pedro Hokama

Fontes

- [*clrs*] Algoritmos: Teoria e Prática (Terceira Edição) Thomas H. Cormen, Charles Eric Leiserson, Ronald Rivest, Ronald L. Rivest e Clifford Stein.
- [*timr*] Algorithms Illuminated Series, Tim Roughgarden

Apresentação Baseada:

- Stanford Algorithms
  `https://www.youtube.com/playlist?list=PLXFMmlk03Dt7Q0xr1PIAriY5623cKiH7V`
  `https://www.youtube.com/playlist?list=PLXFMmlk03Dt5EMI2s2WQBsLsZl7A5HEK6`
- Conjunto de Slides dos Professores Cid C. de Souza, Cândida N. da Silva, Orlando Lee, Pedro J. de Rezende
- Conjunto de Slides do Professores Cid C. de Souza para a disciplina MO420

Qualquer erro é de minha responsabilidade.

Revisão(zinha) de Probabilidade - parte2

Probabilidade Condicional

Na primeira revisão de probabilidade vimos:

- Espaço Amostral
- Eventos
- Variáveis Aleatórias
- Esperança
- Linearidade da Esperança

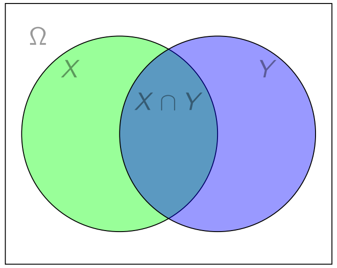
Hoje veremos:

- Probabilidade condicional
- Independência de Eventos e Variáveis Aleatórias

Probabilidade Condicional

Entender a probabilidade de um evento dado um segundo evento.

- Considere os eventos $X, Y \subseteq \Omega$.



- Então a probabilidade de X dado Y é:

$$Pr[X|Y] = \frac{Pr[X \cap Y]}{Pr[Y]}$$

- Suponha que você jogue 2 dados, qual a probabilidade de pelo menos um dado ser 1 dado que a soma deles é 7?
    - 1/36
    - 1/6
    - 1/3
    - 1/2
- X = pelo menos um dado é um 1
- Y = a soma de dois dados é 7
- $Y = \{(1,6),(2,5),(3,4),(4,3),(5,2),(6,1)\}$
- $X \cap Y = \{(1,6),(6,1)\}$

x	2	3	4	5	6	7	8	9	10	11	12
						⚀⚅					
					⚀⚄	⚅⚀	⚁⚅				
S'				⚀⚃	⚄⚀	⚁⚄	⚅⚁	⚂⚅			
			⚀⚂	⚀⚀	⚁⚃	⚄⚁	⚂⚄	⚅⚂	⚃⚅		
		⚀⚁	⚂⚀	⚁⚂	⚃⚁	⚂⚃	⚄⚂	⚃⚄	⚅⚃	⚄⚅	
	⚀⚀	⚁⚀	⚁⚁	⚂⚁	⚂⚂	⚃⚂	⚃⚃	⚄⚃	⚄⚄	⚅⚄	⚅⚅
$\lvert S'\rvert$	1	2	3	4	5	6	5	4	3	2	1

$$
\begin{aligned}
Pr[\text{um dado ser 1}|\text{a soma é sete}] &= \frac{Pr[\text{um dado ser 1 e a soma ser sete}]}{Pr[\text{soma ser sete}]} \\
&= \frac{2/36}{6/36} = \frac{2}{36}\frac{36}{6} = \frac{2}{6} = \frac{1}{3}
\end{aligned}
$$

Independência de Eventos

Definição

Eventos $X, Y \subseteq \Omega$ são independentes se e somente se

$$Pr[X \cap Y] = Pr[X].Pr[Y]$$

- Uma definição equivalente é $P[X|Y] = Pr[X]$

Independência de Variáveis Aleatórias

Definição

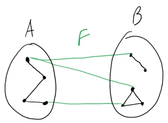
Variáveis aleatórias A e B definidas para um mesmo Ω são independentes se e somente os eventos $A = a$ e $B = b$ são independentes para todo a e b

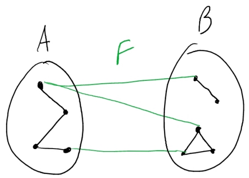
- Uma definição equivalente é $P[A = a \text{ e } B = b] = Pr[A = a] \cdot Pr[B = b]$
- Se as variáveis são independentes $E[A \cdot B] = E[A] \cdot E[B]$ (diferentemente da linearidade da esperança, essa propriedade só se aplica a variáveis independentes)

Algoritmo Aleatorizado de Contração

Qual a probabilidade de sucesso?

- Queremos encontrar um limitante inferior para essa probabilidade. Ou seja queremos mostrar que a probabilidade do algoritmo encontrar um Corte Mínimo não é menor do que um determinado valor.
- Considere um Grafo $G = (V, E)$ com n vértices e m arestas.
- Considere o corte mínimo (A, B), queremos encontrar esse corte! (Podem existir outros cortes mínimos, mas vamos considerar que estamos interessados somente nesse, já que queremos um limitante inferior)
- Seja F o conjunto de arestas de corte em (A, B) e $|F| = k$.





- Se uma aresta de F for escolhida durante o algoritmo, um vértice de A e um vértice de B serão fusionados, causando um corte diferente, e portanto o algoritmo falhará.
- Já se nas $n-2$ iterações apenas arestas com os dois extremos em A ou os dois extremos em B forem selecionadas, o algoritmo vai ser bem sucedido.
- $Pr[\text{devolver } (A, B)] = Pr[\text{não contrair uma aresta de } F]$

- Queremos saber a probabilidade de nenhuma aresta em F ser contraída no algoritmo.
- Seja S_i o evento de que uma aresta de F foi contraída na iteração i
- Então $\neg S_i$ é o evento de nenhuma aresta de F ser contraída na iteração i
- A probabilidade do nosso algoritmo funcionar será:

$$\boxed{Pr[\neg S_1 \cap \neg S_2 \cap \neg S_3 \cap \neg S_4 \cap \ldots \cap \neg S_{n-3} \cap \neg S_{n-2}]}$$

- Qual é a probabilidade de uma aresta do corte (A, B) ser escolhida na primeira iteração? Sendo n o número de vértices, m o número de arestas e k o número de arestas do corte.
  - k/n
  - k/m
  - k/n^2
  - n/m
- Para as próximas iterações será complicado encontrar essa probabilidade em termos do número de arestas, pois o número de aresta varia de maneira imprevisível.
- Então será útil encontrar um limite para essa probabilidade em termos do número de vértices. Já que esse número se comporta bem: A cada iteração diminuímos em 1 o número de vértices.

Definição

O **grau** de um vértice v é o número de arestas incidentes em v. Normalmente denotado por $\delta(v)$.

- O grau de qualquer vértice em V é pelo menos k. (Do contrário $(\{v\}, V - \{v\})$ seria um corte melhor do que (A, B)).
- Note que

$$\sum_{v \in V} \delta(v) = 2m$$

já que cada aresta contribui em 2 para a soma total dos graus.

$$m = \frac{\sum_{v \in V} \delta(v)}{2} \geq \frac{\sum_{v \in V} k}{2}$$
$$m \geq \frac{kn}{2}$$

Como $m \geq \frac{kn}{2}$, então:

$$Pr[S_1] = \frac{k}{m} \leq \frac{k}{kn/2} = k \cdot \frac{2}{kn} = \frac{2}{n}$$
$$Pr[S_1] \leq \frac{2}{n}$$

- Então a probabilidade do algoritmo falhar na primeira iteração é menor que $2/n$
- Note também que a probabilidade de não falhar é $Pr[\neg S_1] \geq \left(1 - \frac{2}{n}\right) = \frac{n-2}{n}$

- Queremos então saber a probabilidade do algoritmo não contrair uma aresta de F na segunda iteração dado que não contraiu na primeira, ou seja:
- Vejamos o complemento, a evento de contrair uma aresta de F na segunda iteração dado que não contraiu na primeira, ou seja:

$$Pr[S_2 | \neg S_1] = \frac{k}{\text{num. arestas restantes}}$$

- Como cada nó restante é um corte, o grau de cada nó é $\geq k$. E portanto o número total de arestas é

$$\text{num. arestas restantes} \geq k(n-1)/2$$

- então

$$Pr[S_2 | \neg S_1] \leq \frac{k}{k(n-1)/2} = \frac{2}{n-1}$$

- Para qualquer iteração i a probabilidade de eu contrair uma aresta de F dado que eu não o fiz nas iterações anteriores é análogo.

$$Pr\left[S_i | \bigcap_{j<i} \neg S_j\right] = \frac{k}{\text{num. arestas restantes}}$$

- Como cada nó restante também é um corte, o grau de cada nó é $\geq k$, o número total de nós é $n - i + 1$. E portanto o número total de arestas é

$$\text{num. arestas restantes} \geq k(n-i+1)/2$$

- então

$$Pr\left[S_i | \bigcap_{j<i} \neg S_j\right] \leq \frac{k}{k(n-i+1)/2} = \frac{2}{n-i+1}$$

$$Pr\left[S_i | \bigcap_{j<i} \neg S_j\right] \leq \frac{2}{n-i+1}$$

- Dessa forma a probabilidade de **Não** contrair uma aresta de F na iteração i dado que também não contraiu nas anteriores é

$$Pr\left[\neg S_i | \bigcap_{j<i} \neg S_j\right] \geq 1 - \frac{2}{n-i+1} = \frac{n-i+1-2}{n-i+1} = \frac{n-i-1}{n-i+1}$$

$$\boxed{Pr[\neg S_1 \cap \neg S_2 \cap \neg S_3 \cap \neg S_4 \cap \ldots \cap \neg S_{n-3} \cap \neg S_{n-2}]}$$

$$Pr[\neg S_1] \cdot Pr[\neg S_2|\neg S_1] \cdot Pr[\neg S_3|\neg S_1 \cap \neg S_2] \cdot Pr[\neg S_4|\neg S_1 \cap \neg S_2 \cap \neg S_3] \ldots$$
$$\ldots Pr[\neg S_{n-3}|\neg S_1 \cap \ldots \cap \neg S_{n-2}] \cdot Pr[\neg S_{n-2}|\neg S_1 \cap \ldots \cap \neg S_{n-3}]$$

$$\geq \frac{n-2}{n} \cdot \frac{n-3}{n-1} \cdot \frac{n-4}{n-2} \cdot \frac{n-5}{n-3} \ldots \frac{4}{6} \cdot \frac{3}{5} \cdot \frac{2}{4} \cdot \frac{1}{3}$$

$$= \frac{\cancel{n-2}}{n} \cdot \frac{\cancel{n-3}}{n-1} \cdot \frac{\cancel{n-4}}{\cancel{n-2}} \cdot \frac{\cancel{n-5}}{\cancel{n-3}} \ldots \frac{\cancel{4}}{\cancel{6}} \cdot \frac{\cancel{3}}{\cancel{5}} \cdot \frac{2}{\cancel{4}} \cdot \frac{1}{\cancel{3}} = \frac{2}{n(n-1)} = \frac{2}{n^2-n}$$
$$= \frac{2}{n^2-n} \geq \frac{2}{2n^2-n} \geq \frac{2}{2n^2} = \frac{1}{n^2}$$

$$\boxed{Pr[\neg S_1 \cap \neg S_2 \cap \neg S_3 \cap \neg S_4 \cap \ldots \cap \neg S_{n-3} \cap \neg S_{n-2}] \geq \frac{1}{n^2}}$$

$$\boxed{Pr[\neg S_1 \cap \neg S_2 \cap \neg S_3 \cap \neg S_4 \cap \ldots \cap \neg S_{n-3} \cap \neg S_{n-2}] \geq \frac{1}{n^2}}$$

- Então a probabilidade do algoritmo funcionar é... muito baixa!
- PORÉM!!! Veja bem. Se você pegasse um corte aleatório a probabilidade dele ser o (A, B) é $\frac{1}{2^n}$

n	$\frac{1}{2^n}$	$\frac{1}{n^2}$
5	$\frac{1}{32}$	$\frac{1}{25}$
10	$\frac{1}{1024}$	$\frac{1}{100}$
100	$\frac{1}{1267650600228229401496703205376}$	$\frac{1}{10000}$

- Podemos fazer um truque para melhorar essa probabilidade. Basta executar o algoritmo várias vezes!

Múltiplas Execuções

- Iremos executar o algoritmo N vezes, e devolver o menor corte encontrado.
- Quantas execuções serão necessárias?
- Seja T_j o evento de que o corte (A, B) seja encontrado na j-ésima tentativa. T_j são independentes.

$$Pr[\text{falhar nas } N \text{ tentativas}] = Pr[\neg T_1 \cap \ldots \cap \neg T_N]$$
$$(ind.) = \prod_{j=1}^{N} Pr[\neg T_j] \leq \left(1 - \frac{1}{n^2}\right)^N$$

$$Pr[\text{falhar nas } N \text{ tentativas}] \leq \left(1 - \frac{1}{n^2}\right)^N$$

- Para qualquer numero real x, $1 + x \leq e^x$

$$\begin{aligned} Pr[\text{falhar nas } N \text{ tentativas}] &\leq \left(1 - \frac{1}{n^2}\right)^N \\ &\leq \left(e^{-1/n^2}\right)^N \\ &= e^{-N/n^2} \\ &= \frac{1}{e^{N/n^2}} \end{aligned}$$

- Para $N = n^2$

$$Pr[\text{falhar nas } n^2 \text{ tentativas}] \leq \frac{1}{e} \approx 37\%$$

- Para $N = n^2 \ln n$

$$Pr[\text{falhar nas } n^2 \ln n \text{ tentativas}] \leq \frac{1}{e^{\ln n}} = \frac{1}{n}$$