Aula 18 - Algoritmos Randomizados

Luís Felipe

UFF

28 de Novembro de 2023

Luís Felipe
28/11/23

Algoritmos Randomizados

- Um experimento é aleatório ou randômico quando seu resultado advém de uma quantidade tão grande de variáveis que torna-se impossível prevê-lo.
  - O lançamento de um dado é um experimento aleatório, há muitas variáveis inviáveis de prever.
- Um algoritmo é randomizado quando em pelo menos um momento alguma de suas decisões é tomada através de um experimento aleatório.
- Um algoritmo clássico (não-randomizado) é determinístico.
- É conveniente imaginar que um algoritmo randomizado lança um dado (com número qualquer de faces) para decidir qual ação será executada.
  - Na prática, o computador usa geradores de números "aleatórios" para executar o papel dos dados.
  - Esses geradores são algoritmos determinísticos que sorteiam um número em um conjunto finito com probabilidade bem próxima da uniforme.

Luís Felipe
28/11/23

Qual o preço que se paga?

- Algoritmos randomizados têm aplicações em diversos campos (criptografia, computação distribuída, teoria dos grafos etc.)
- Motivo: eles são, em geral, mais simples ou mais eficientes que os determinísticos respectivos.
- Mas a aleatoriedade cobra um preço: a incerteza
  - Será que essa resposta está certa?
    - Vamos ver que é possível conhecermos quão provável é a exibição de uma resposta incorreta.
  - Será que esse algoritmo para em tempo hábil?
    - Vamos ver que nem a pior entrada é ruim o suficiente para forçar o algoritmo a executar um número excessivo de passos

Luís Felipe
28/11/23

Monte Carlo e Las Vegas

Existem dois grandes grupos de algoritmos randomizados:

- Monte Carlo: incerteza na corretude da resposta
- Las Vegas: incerteza no tempo de execução

Qual usar? Dependerá do problema.

- Se tempo é deterministicamente limitado em função do tamanho da entrada, então Monte Carlo
- Se certeza na resposta é necessário, então Las Vegas.

Luís Felipe

28/11/23

Monte Carlo

- Nem sempre dão a resposta certa
- Associados a problemas de decisão
- Divididos em dois tipos: de erro unilateral e de erro bilateral
  - Erro unilateral:
    - Baseados-no-SIM: Fornece certificado para o SIM. Nunca erra quando a reposta encontrada é SIM
    - Baseados-no-NÃO: Fornece certificado para o NÃO. Nunca erra quando a reposta encontrada é NÃO
  - Erro bilateral: Podem retornar tanto SIM quanto NÃO incorretamente
- A unilateralidade do erro de um MC permite refinar o grau de confiança da resposta a níveis tão bons quanto queiramos.

Luís Felipe
28/11/23

Exemplo:

1. Identidade de polinômios
   - $F(x) = (x - a_1)(x - a_2)\ldots(x - a_d)$
   - $G(x) = b_d x^d + b_{d-1} x^{d-1} + \ldots + b_1 x + b_0$
   - $F(x) = G(x)$?

Alg. determinístico	Alg. Monte Carlo
Transforma $F(x)$ Compara os coef. de $F(x)$ e $G(x)$ Retorna SIM, caso os coef. sejam iguais Retorna NÃO, caso contrário	

Luís Felipe
28/11/23

Exemplo:

1. Identidade de polinômios
   - $F(x) = (x - a_1)(x - a_2)\ldots(x - a_d)$
   - $G(x) = b_d x^d + b_{d-1}x^{d-1} + \ldots + b_1 x + b_0$
   - $F(x) = G(x)$?

Alg. determinístico	Alg. Monte Carlo
Transforma $F(x)$ Compara os coef. de $F(x)$ e $G(x)$ Retorna SIM, caso os coef. sejam iguais Retorna NÃO, caso contrário	Sorteia w entre 1 e 100 d Se $F(w) = G(w)$, então retorna SIM Caso contrário, retorna NÃO

Luís Felipe
28/11/23

Exemplo:

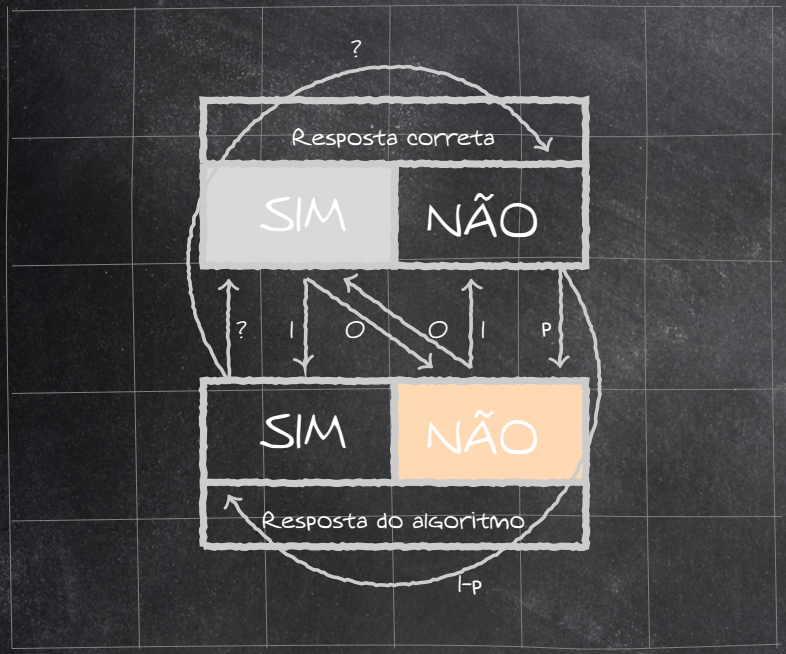
1. Identidade de polinômios
   - $F(x) = (x - a_1)(x - a_2)\dots(x - a_d)$
   - $G(x) = b_d x^d + b_{d-1} x^{d-1} + \dots + b_1 x + b_0$
   - $F(x) = G(x)$?

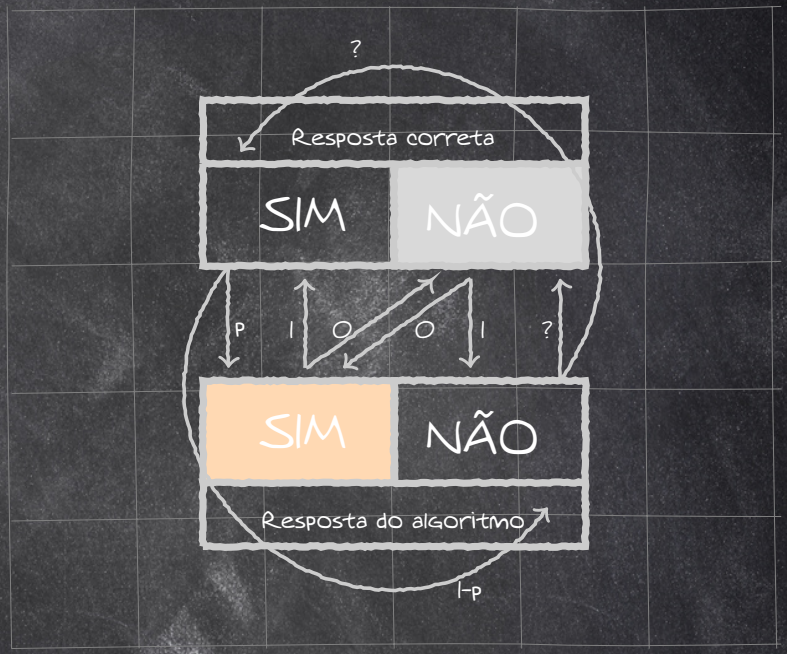
Alg. determinístico $O(d^2)$	Alg. Monte Carlo $O(d)$
Transforma $F(x)$ Compara os coef. de $F(x)$ e $G(x)$ Retorna SIM, se os coef. sejam iguais Retorna NÃO, caso contrário	Sorteia w entre 1 e 100 d Se $F(w) = G(w)$, então retorna SIM Caso contrário, retorna NÃO

Este algoritmo de Monte Carlo é Baseado-no-NÃO

Luís Felipe
28/11/23

Esquemas Baseados -no-NÃO / -no-SIM





Obs: Probabilidade da Resposta Correta ser NÃO dado que o Algoritmo retorna NÃO é igual a 1.

Obs: Probabilidade da Resposta Correta ser SIM dado que o Algoritmo retorna SIM é igual a 1.

Seta de X para Y representa a probabilidade de Y dado X.

Luís Felipe

28/11/23

Recordar é viver... Probabilidade

A função de probabilidade P precisa atender às condições:

1. Para todo evento $A \subseteq \Omega$, $0 \leq P(A) \leq 1$, onde Ω é o espaço amostral
2. $P(\Omega) = 1$
3. Para eventos $A_1, A_2, \ldots$, dois-a-dois disjuntos, $P(\bigcup_i A_i) = \sum_i P(A_i)$

Luís Felipe

28/11/23

Recordar é viver... PIE

Sejam $A_1, A_2, \ldots$ eventos arbitrários. Então

$$\begin{aligned} P(\bigcup_i A_i) \quad = \quad & \sum_i P(A_i) \\ & - \sum_{i<j} P(A_i \cap A_j) \\ & + \sum_{i<j<k} P(A_i \cap A_j \cap A_k) \\ & - \ldots + (-1)^{l+1} \sum_{i_1<i_2<\ldots<i_l} P(\bigcap_{r=1}^{l} A_{i_r}) \\ & + \ldots \end{aligned}$$

Assim, um limite para a união é dado por:

$$P(\bigcup_i A_i) \leq \sum_i P(A_i)$$

Obs.: Nem sempre temos informações tão completas das interseções dos eventos então ficamos satisfeitos marjorando superiormente a probabilidade.

Luís Felipe

28/11/23

Recordar é viver... Probabilidade Condicional

A probabilidade condicional de um evento A dado um evento B, denotada por $P(A|B)$, corresponde à probabilidade de que o resultado do experimento aleatório pertença a A sabendo-se que também pertence a B.

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(B|A)P(A)}{P(B)}$$

OBS.: Quando os eventos são independentes entre si, $P(A|B) = P(A)$.

Da definição de probabilidade condicional, concluímos que:

$$P(\bigcap_{i=1}^{n} A_i) = \Pi_{i=1}^{n} P(A_i | \bigcap_{j<i} A_j)$$

Se os eventos forem dois-a-dois independentes, então:

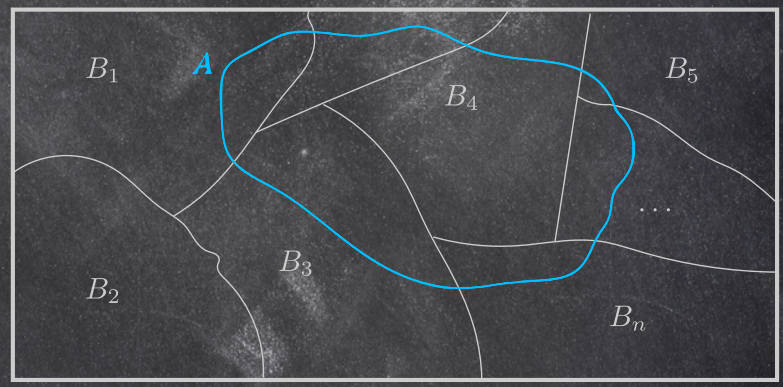
$$P(\bigcap_{i=1}^{n} A_i) = \Pi_{i=1}^{n} P(A_i)$$

Luís Felipe

28/11/23

Recordar é viver... Regra de Bayes

Seja $\Omega = \{B_1, B_2, \ldots, B_n\}$ o particionamento do espaço amostral Ω. Qual a probabilidade de um evento A qualquer acontecer?



$$P(A) = P(A \cap \bigcup_i B_i) = \sum_{i=1}^{n} P(A \cap B_i) = \sum_{i=1}^{n} P(A|B_i)P(B_i)$$

Regra de Bayes $$P(B_k|A) = \frac{P(B_k \cap A)}{P(A)} = \frac{P(A|B_k)P(B_k)}{\sum_{i=1}^{n} P(A|B_i)P(B_i)}$$

Luís Felipe

28/11/23

Uma aplicação do Teorema de Bayes

Problema: Um fabricante de sorvetes recebe 20% de todo o leite da fazenda F_1, 30% da fazenda F_2 e 50% da fazenda F_3.

Foram encontras adulterações: 20% dos leites de F_1, 5% dos leites de F_2 e 2% dos leites de F_3.

Seja A o evento "leite adulterado". F_1, F_2 e F_3 formam uma partição do espaço amostral de onde vem o leite.

Assim: $P(A|F_1) = 0,20$, $P(A|F_2) = 0,05$ e $P(A|F_3) = 0,02$

Além disso, $A = (A \cap F_1) \cup (A \cap F_2) \cup (A \cap F_3)$

Pergunta: Qual a probabilidade do leite ser da fazenda F_1 dado que está adulterado?

$$P(F_1|A) = \frac{P(F_1 \cap A)}{P(A)} = \frac{P(A|F_1)P(F_1)}{P(A|F_1)P(F_1)+P(A|F_2)P(F_2)+P(A|F_3)P(F_3)}$$

$$= \frac{0,2 \times 0,2}{(0,2 \times 0,2)+(0,3 \times 0,05)+(0,5 \times 0,02)} = 0,615$$

Luís Felipe
28/11/23

Voltando ao exemplo

- O algoritmo sempre acerta a resposta quando os polinômios são idênticos ou quando retorna NÃO
- Qual a probabilidade do algoritmo errar a resposta? O que significa errar neste caso?
  - O algoritmo responde SIM e a resposta correta é NÃO
  - Isso acontece quando o número sorteado é raiz de $F(x) - G(x)$ mas os polinômios não são os mesmos. Qual o grau de $F(x) - G(x)$? No máximo d
    - Pelo Teorema Fundamental da Álgebra, esse polinômio tem no máximo d raízes distintas
    - Assim, a probabilidade de erro é, no máximo, a probabilidade de tomar alguma dessas raízes
    - Ou seja, $\varepsilon \leq \frac{d}{100d} = \frac{1}{100} = 1\%$
    - Logo, a probabilidade de acerto é de pelo menos 99%

Luís Felipe

28/11/23

Notação Monte Carlo

Sejam os eventos associados a uma execução do algoritmo para uma determinada instância de um problema de decisão:

- C_N : a resposta correta é NÃO
- C_S : a resposta correta é SIM
- A_N : o algoritmo responde NÃO
- A_S : o algoritmo responde SIM

Baseado-no-NÃO: $P(C_N|A_N) = 1 \leftrightarrow P(A_S|C_S) = 1$

Luís Felipe

28/11/23

Probabilidade do erro

$$
\begin{aligned}
P(\text{erro}) &= P([C_N \cap A_S] \cup [C_S \cap A_N]) \\
&= P(C_N \cap A_S) + P(C_S \cap A_N) \\
&= P(C_N)P(A_S|C_N) + P(C_S)P(A_N|C_S) \\
&= P(C_N)P(A_S|C_N) + P(C_S)0 \\
&= P(C_N)\varepsilon \\
P(\varepsilon) &\leq \varepsilon \\
&\downarrow \\
&P(\text{acerto}) \geq p = 1 - \varepsilon
\end{aligned}
$$

Luís Felipe

28/11/23

Refinamento

- A um algoritmo de Monte Carlo baseado-no-NÃO
- A erra com probabilidade menor ou igual a ε
- I é uma instância qualquer do problema de decisão
- Plano: Vamos executar A várias vezes de maneira independente até que o NÃO seja respondido ou por t vezes
- Quando ele erra? Quando a resposta certa é NÃO e ele não consegue encontrar o certificado em t execuções
- Assim, $P(\text{erro}) = P(\text{erro}_1, \text{erro}_2, \ldots, \text{erro}_t) \leq \varepsilon^t$
- Consequência: A probabilidade de erro decresce exponencialmente com o aumento de número de repetições independentes do algoritmo

Luís Felipe

28/11/23

Las Vegas

- Sempre dão a resposta certa
- O tempo computacional de um algoritmo de Las Vegas é uma variável aleatória
  - como tal, está completamente definido por seu conjunto de momentos
  - o tempo computacional de um algoritmo de Las Vegas é avaliado em termos de seu valor esperado, e talvez variância, desvio padrão, etc.

Luís Felipe
28/11/23

Exemplo:

1. Busca de elemento em lista com repetições
   - Desejamos localizar um algarismo qualquer (digamos, o 9) numa lista de tamanho n
   - A lista contém todos os algarismos de 0 a 9 distribuídos em iguais quantidades
     - isto é, 1/10 de suas posições apresentam o algarismo 0, 1/10 de suas posições apresentam o algarismo 1 e assim por diante.
   - Nada se sabe sobre a localização dos elementos.

Luís Felipe

28/11/23

Soluções: Estratégias determinísticas

1. Examine uma a uma todas as posições da lista, a partir da primeira, até encontrar o primeiro 9;
2. Examine uma a uma todas as posições da lista, a partir da última e caminhando de trás para diante, até encontrar o primeiro 9;
3. Examine primeiro todas as posições ímpares da lista, isto é, a primeira, depois a terceira, quinta etc. Depois (se nenhum 9 tiver ainda sido encontrado, evidentemente) venha voltando pelas posições pares de trás para diante;
4. Divida a lista em k sublistas de tamanho n/k cada: os primeiros n/k elementos irão para a primeira sublista, os n/k elementos seguintes irão para a segunda sublista e assim por diante. Examine agora o primeiro elemento de cada sublista, em seguida o segundo elemento de cada sublista, em seguida o terceiro etc. até encontrar um 9.

Luís Felipe
28/11/23

Conseguimos entradas desfavoráveis?

- Qualquer que seja a estratégia adotada, sempre há entradas que exigirão do algoritmo um tempo "ruim" (linear no tamanho da entrada, neste exemplo).
  - O algoritmo determinístico pode ser constantemente levado a ter um desempenho lento;
  - Depende da aplicação e da distribuição das instâncias de entrada, por força de algum agente externo, malicioso ou não, ou ainda que intermitentemente.

Luís Felipe

28/11/23

Solução: Estratégia randomizada

1. Escolha, aleatória e uniformemente, uma posição qualquer, das n possíveis. Verifique-a. Repita até encontrar um 9.

OBS.: Esta estratégia irá encontrar o 9. Porém, como saber o número de verificações até encontrar o primeiro 9?

Continuaremos na próxima aula!