Instituto Superior de Engenharia de Lisboa

Licenciatura em Engenharia Informática e Multimédia
Licenciatura em Engenharia Informática e de Computadores

Codificação de Sinais Multimédia

2º Semestre de 2024/2025

A Considere o seguinte problema: O António atira uma moeda viciada N vezes, obtendo uma sequência de caras e coroas. Assumimos que a probabilidade de sair cara é f_H; não conhecemos f_H. O número de vezes que ocorreram caras em N lançamentos designa-se n_H.

Assumindo um *prior* uniforme sobre f_H, apresente um gráfico da distribuição posterior de f_H e calcule a probabilidade de o lançamento $N+1$ ser cara para:

(a) $N = 3$ e $n_H = 0$; (b) $N = 3$ e $n_H = 2$; (c) $N = 10$ e $n_H = 3$; (d) $N = 300$ e $n_H = 29$.

B Considere que dispõe de um conjunto de imagens. Dispõe também de um programa de computador que faz compressão de imagens recorrendo a 3 algoritmos possíveis (um de cada vez que o programa é executado): A, B, C.

O programa, de cada vez que é aplicado a uma imagem, escolhe de forma aleatória qual o algoritmo que vai utilizar. A probabilidade *a priori* de escolha de cada um dos algoritmos é dada por: $P(A) = 0.4$, $P(B) = 0.35$, $P(A) = 0.25$.

A taxa de compressão de cada um dos algoritmos tem também um caráter aleatório[1] e pode ser classificada em cinco níveis possíveis: 0, 1, 2, 3, 4 de acordo com o nível de compressão (0 menor compressão, 4 maior compressão).

A probabilidade condicional de cada um desses níveis de compressão dado o algoritmo é representada por:

Nível de compressão (NC)	$P(NC\|A)$	$P(NC\|B)$	$P(NC\|C)$
0	0.05	0.10	0.20
1	0.15	0.20	0.25
2	0.40	0.40	0.30
3	0.30	0.20	0.15
4	0.10	0.10	0.10

(a) Defina o *prior* `P_algoritmo` como um array de comprimento 3. Crie uma matriz de verosimilhança `P_NC_dado_algoritmo`. Calcule a distribuição conjunta `P_conjunta`. Determine a distribuição marginal dos níveis de compressão.

(b) Implemente uma função `classifica_algoritmo(NC)` que tem como output a distribuição de probabilidades *a posteriori* para cada algoritmo dado o nível de compressão observado. Utilize esta função para classificar o algoritmo utilizado sendo que o nível de compressão observado foi NC = 3.

(c) Determine o valor mais verosímil de nível de compressão para cada algoritmo de compressão (Máxima Verosimilhança). Determine o algoritmo mais provável para cada nível de compressão NC observado (*Maximum a posteriori*).

(d) Determine o valor esperado do nível de compressão NC para cada algoritmo. Determine também a variância.

[1] Pois depende de diversas características das imagens e estas são escolhidas de forma aleatória de entre um conjunto muito grande.

(e) Gere 50000 escolhas aleatórias do algoritmo de compressão com base na função de verosimilhança e no *prior*. Faça uma estimativa empírica de $P(\text{algoritmo}|\text{NC} = 2)$ a partir dos dados gerados. Compare os valores obtidos com o valor teórico.

Requisitos/Dicas

- Validar o axioma das probabilidades somarem 1 (ex.: `assert np.allclose(np.sum(P_conjunto), 1.0)`).
- Deve utilizar arrays de NumPy.
- Simulação: para fazer *sampling* poderá achar útil `np.random.choice`

C Considere uma sequência de N lançamentos de uma moeda viciada $\mathbf{x} = (x_1, \ldots, x_N)$, onde $x_n \in \{0, 1\}$ com probabilidades $p_0 = 0.9$ e $p_1 = 0.1$. Designando por $r(x)$ o número de 1s em $\mathbf{x}$ então: $P(\mathbf{x}) = p_0^{N-r(\mathbf{x})} p_1^{r(\mathbf{x})}$.

(a) Para $N = 4$ e $N = 10$ reproduza o gráfico de $H_\delta(X^N)$ em função de δ (Figuras 4 e 5 do deck de slides de codificação de fonte ou Figuras 4.7 e 4.8 em [1], `http://www.inference.phy.cam.ac.uk/mackay/itila/book.html`).

(b) Reproduza o gráfico de $\frac{1}{N} H_\delta(X^N)$ em função de δ da Figura 6 do deck de slides de codificação de fonte, ou Figura 4.9 em [1]. Para tal considere $N = 10, 210, 410, 610, 810, 1010$. Este gráfico é especialmente interessante, pois corresponde à verificação empírica do Teorema da Codificação de Fonte de Shannon.

Tenha em atenção que para $N = 210$, ou superior, não é possível gerar e ter em memória todas as sequências binárias de comprimento N. Para vermos porquê basta ter em conta que $2^{10} \approx 10^3$ e que[2] $2^{210} = (2^{10})^{21} \approx (10^3)^{21} = 10^{63}$. Tal número é astronómico (o número de átomos no universo é aproximadamente 10^{80}). Assim terá que fazer amostragem. Obtenha, para cada valor de N, 1000 amostras (sequências binárias de comprimento N) aleatoriamente obtidas. Deverá gerar amostras com cada um dos possíveis números de 1s (de 0 a N). O número de amostras de cada um destes tipos deverá ser aproximadamente igual.

References

[1] David J. C. MacKay. Information theory, inference, and learning algorithms, 2003.

[2] O número de sequências binárias distintas, de comprimento 210 é 2^{210}.