

Sistemas Multimédia

Inverno 2021/2022

Trabalho prático I

- 1 – Entendendo o conteúdo de um ficheiro como uma sequência de símbolos (bytes) produzida por uma fonte discreta com alfabeto $\{0, 1, \dots, 255\}$, desenvolva uma aplicação que recebendo o nome do ficheiro:
 - a) Estime a entropia de primeira ordem e determine a maior frequência de ocorrência;
 - b) Determine o menor número de símbolos que faz com a frequência de ocorrência de símbolos desse subconjunto seja igual ou superior à dos restantes símbolos;
 - c) Estime a entropia de segunda ordem;
 - d) Com a aplicação desenvolvida analise os ficheiros do corpus e comente as estimativas da entropia.
- 2 – Para cada ficheiro, admitindo modelo de Markov de 1.^a ordem, estime a entropia e mostre o menor e o maior valor da entropia de cada estado. Analise os ficheiros do corpus e compare com as estimativas de primeira e de segunda ordem.
- 3 – Considere uma fonte com memória (modelo de Markov de 1.^a ordem), alfabeto $\{0, 1, \dots, 255\}$ e probabilidades de transição $\{(1-p)/4, (1-p)/4, p/3, p/3, p/3, (1-p)/4, (1-p)/4\}$ do estado s para os estados $\{s-3, s-2, s-1, s, s+1, s+2, s+3\} \bmod 256$, respetivamente.
 - a) Calcule a entropia da fonte. Obtenha sequências de acordo com esse modelo e analise, para valores de p entre 0 e 1, a compressão obtida com uma aplicação de compressão.
 - b) Proponha uma forma eficiente de codificação de sequências geradas, determine o comprimento médio desse código e compare com os resultados da alínea anterior.
- 4 – Considere o alfabeto $\{A, B, C, D, E\}$:
 - a) Identifique a estrutura das árvores binárias que estão na origem de códigos de Huffman canónicos para esse alfabeto e, para cada uma dessas árvores, determine o número de códigos diferentes existentes e o comprimento médio do código ideal;
 - b) Desenhe um código eficiente para representar cada código canónico e compare com a utilização de código de comprimento constante para representar o comprimento de cada palavra de código.
- 5 – Considere sequências de números de 0 a 16 (alfabeto com 17 símbolos) que indicam o número de '1' em valores a 16 bit gerados aleatoriamente com distribuição uniforme. Determine a entropia da fonte. Implemente um simulador da fonte e analise a compressão obtida usando uma aplicação de compressão.
- 6 – Pretende-se verificar se existe vantagem em usar predição na codificação de ficheiros, admitindo alfabeto com 256 símbolos. Considerando o erro de predição $e(n) = x(n) - P_i(x(n))$ e os preditores $P_0(x(n)) = 0$, $P_1(x(n)) = x(n-1)$, $P_2(x(n)) = 2x(n-1) - x(n-2)$, $P_3(x(n)) = x(n-1) + x(n-2) - x(n-3)$ e $P_4(x(n)) = x(n-2)$ analise o erro usando vários ficheiros e, tendo em vista maximizar a compressão, estabeleça, justificando, o critério para escolher o melhor preditor para cada ficheiro. Em cada preditor até haver o número de símbolos necessário para a predição $P_i(x(n)) = 0$.
- 7 – Pretende-se implementar um gerador de mensagens curtas com comprimento máximo L , usando um ficheiro com N bytes (alfabeto com 256 símbolos) e $N \gg L$. Cada mensagem é gerada escolhendo aleatoriamente, com distribuição uniforme, o início da mensagem no ficheiro e emitindo uma sequência de símbolos a partir desse ponto.
 - a) Considerando geração de mensagens com comprimento constante ($L=128$), determine o número médio de bits mínimo necessário para codificar cada símbolo da mensagem. Justifique as aproximações consideradas.
 - b) Admitindo codificação de Huffman adaptativa das mensagens, determine o menor e o maior número de bits necessário para codificar uma mensagem. Justifique as aproximações.
 - c) Considerando agora geração de mensagens com comprimento com distribuição uniforme entre $L/2$ e $L=256$, determine o número médio de bits mínimo necessário para codificar cada símbolo da mensagem. Justifique as aproximações.
 - d) Compare os resultados anteriores com a compressão de sequências de mensagens geradas usando um algoritmo de compressão baseado em dicionário (e.g., DEFLATE).