

Université Badji-Mokhtar. Annaba
Faculté de Technologie
Département d'Informatique



Chapitre 2

Méthodes de Compression Statistiques

S3- 2025/2026



Introduction

- **Compression statistique:**
 - ✓ s'appuie sur l'analyse des données pour déterminer la fréquence à laquelle chaque symbole apparaît, puis attribue des codes binaires plus courts aux symboles les plus fréquents et des codes plus longs aux symboles les moins fréquents.
 - ✓ Utilisé dans les applications où les fréquences des symboles sont très différentes (JPEG, MPEG)
 - ✓ **Exemples:** Codage de Huffman, Codage de Shannon-Fano.
- 

Codage de Shannon-Fano

- ✓ Le codage de Shannon-Fano est un algorithme de compression de données sans perte.
- ✓ Développé par Claude Shannon (MIT) et Robert Fano.
- ✓ Il consiste à attribuer un code à chaque symbole en fonction de leurs probabilités d'occurrence.
- ✓ C'est un codage statistique de longueur variable.
- ✓ Exemple d'utilisation: format ZIP.

Codage de Shannon-Fano

- L'algorithme est le suivant :

1. On classe les symboles par ordre de probabilités décroissantes,
2. On partage l'ensemble des symboles en deux sous-ensembles, les 2 sous-ensembles devant être de probabilités aussi proches que possible,
3. On attribue à chaque sous-ensemble l'état 0 ou 1,
4. On partage chaque sous-ensemble en deux sous-ensembles de probabilités aussi proches que possible, on attribue à chaque nouveau sous-ensemble l'état 0 ou 1, etc.

Codage de Shannon-Fano

- Exemple:

symboles	p_i		c_i
A	0.4		
B	0.18		
C	0.1		
D	0.1		
E	0.07		
F	0.06		
G	0.05		
H	0.04		

Codage de Shannon-Fano

- Exemple:

symboles	p_i			c_i
A	0.4	0		
B	0.18			
C	0.1	1		
D	0.1			
E	0.07			
F	0.06			
G	0.05			
H	0.04			

Codage de Shannon-Fano

■ Exemple:

symboles	p_i			c_i
A	0.4	0	0	
B	0.18		1	
C	0.1	1		
D	0.1			
E	0.07			
F	0.06			
G	0.05			
H	0.04			

Codage de Shannon-Fano

- Exemple:

symboles	p_i				c_i
A	0.4	0	0		
B	0.18		1		
C	0.1	1	0		
D	0.1				
E	0.07		1		
F	0.06				
G	0.05				
H	0.04				

Codage de Shannon-Fano

■ Exemple:

symboles	p_i				c_i
A	0.4	0	0		
B	0.18		1		
C	0.1	0	0		
D	0.1		1		
E	0.07	1	1		
F	0.06				
G	0.05				
H	0.04				

Codage de Shannon-Fano

- Exemple:

symboles	p_i				c_i	
A	0.4	0	0			
B	0.18		1			
C	0.1	0	0			
D	0.1		1			
E	0.07	1	0		0	
F	0.06				0	
G	0.05		1		1	
H	0.04				1	

Codage de Shannon-Fano

- Exemple:

symboles	p_i					c_i	
A	0.4	0	0				
B	0.18		1				
C	0.1	1	0	0			
D	0.1			1			
E	0.07		0	0			
F	0.06			1			
G	0.05		1				
H	0.04						

Codage de Shannon-Fano

- Exemple:

symboles	p_i					c_i		
A	0.4	0	0					
B	0.18		1					
C	0.1	1	0	0				
D	0.1			1				
E	0.07		1	0	0			
F	0.06				1			
G	0.05			1	1	0		
H	0.04					1		

Codage de Shannon-Fano

- Exemple:

symboles	p_i					c_i		
A	0.4	0	0			00		
B	0.18		1			01		
C	0.1	1	0	0			100	
D	0.1			1			101	
E	0.07		1	0	0			1100
F	0.06				1			1101
G	0.05	1		1	0			1110
H	0.04				1			1111

Codage de Shannon-Fano

- **Exemple:**

- ✓ **La longueur moyenne de code:**

$$L_m = 0.22 * 2 + 0.2 * 3 + 0.22 * 4 = 2.64 \text{ bits/symbole}$$

- ✓ **L'entropie:** $H = 2.55 \text{ bits/symbole}$

- ✓ **L'efficacité du codage est** $E = 2.55 / 2.64 \approx 96.6\%$



Codage de Shannon-Fano

- **Exemple décompression:** 11101100001101101100111101



Codage de Shannon-Fano

- Exemple décompression: **GEAFDCHB**
- $T=26/64=0.41$
- $G=1-T=59\%$

Codage de Shannon-Fano

✓ Avantages:

- Algorithme simple à implémenter avec des performances élevées.
- Plus la probabilité d'occurrence est élevée, plus la longueur du code est courte.

✓ Limites:

- Nécessite une connaissance préalable des probabilités des symboles.
- Moins optimal que le codage de Huffman

Codage de Huffman

- ✓ Le codage de Huffman est un algorithme de compression de données sans perte
- ✓ Développé en 1952 par David A. Huffman
- ✓ Permet de déterminer des codes à longueurs variables pour une source donnée
- ✓ Associer aux symboles les moins fréquents plus de bits, et inversement (c.-à-d. mettre les lettres les moins fréquentes les plus en bas de l'arbre de codage.)
- ✓ Très proche du codage de Shannon-Fano, mais les codes de Huffman sont optimaux
- ✓ Exemple d'utilisation: format ZIP, PNG, JPEG, MP3, etc.

Codage de Huffman

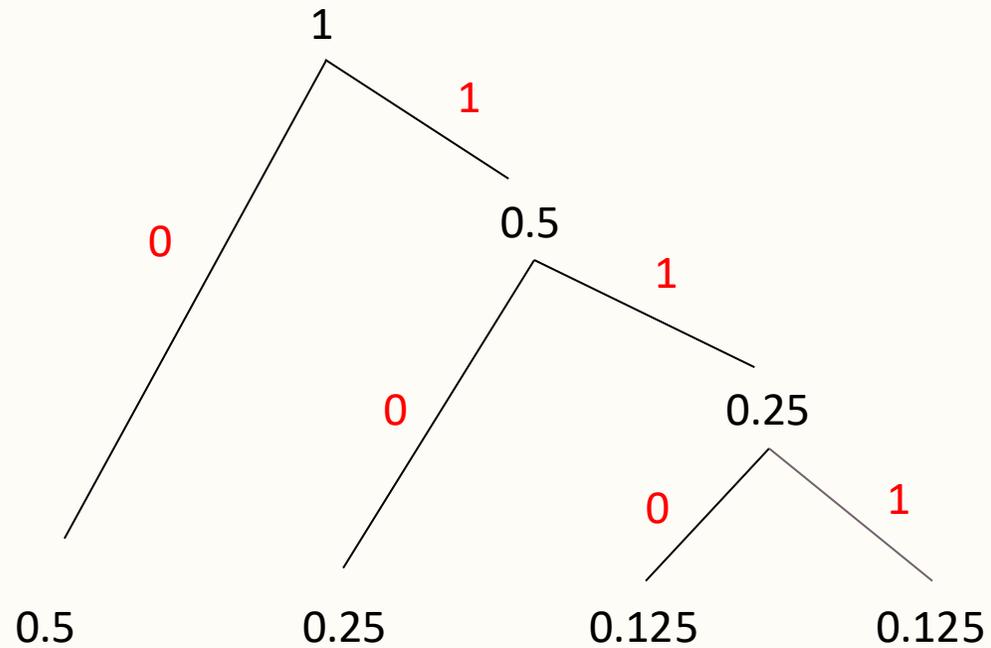
- L'algorithme est le suivant :

1. Ranger les symboles par ordre décroissant de fréquences
2. Combiner les 2 symboles les moins probables p_i et p_j
3. Former, à partir de ces 2 symboles, un nœud dont la probabilité est la somme $p_i + p_j$
4. Ignorer les deux nœuds d'origine
5. Revenir à l'étape 2 jusqu'à obtenir un seul nœud de probabilité 1

Codage de Huffman

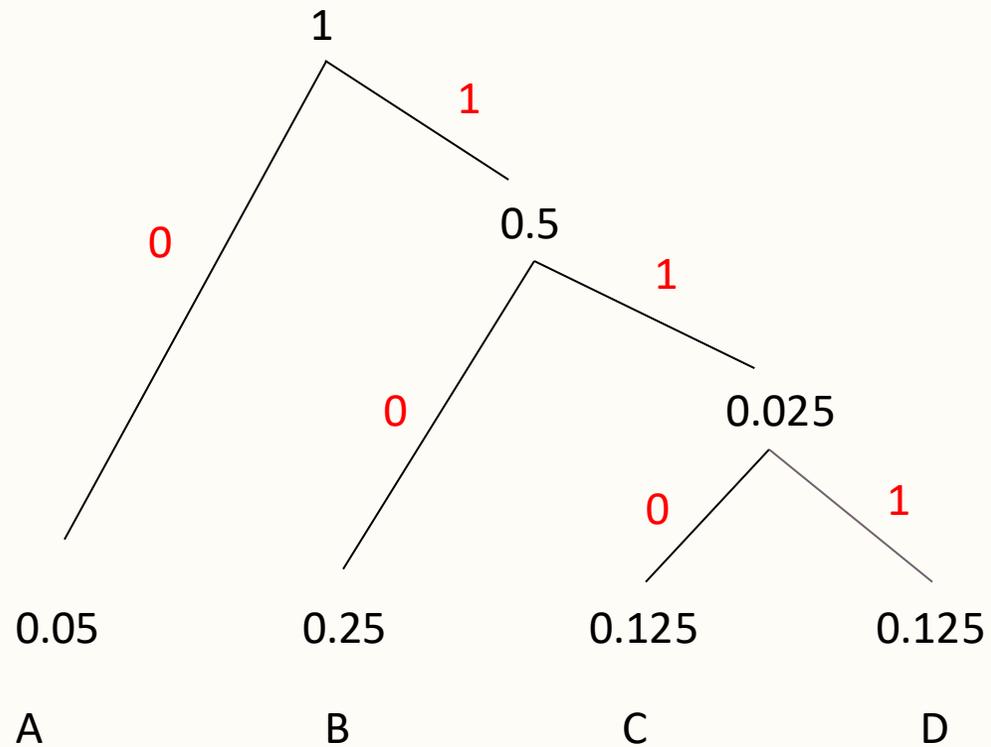
■ Exemple:

Lettres	A	B	C	D
Probabilité	0.5	0.25	0.125	0.125
Code	0	10	110	111



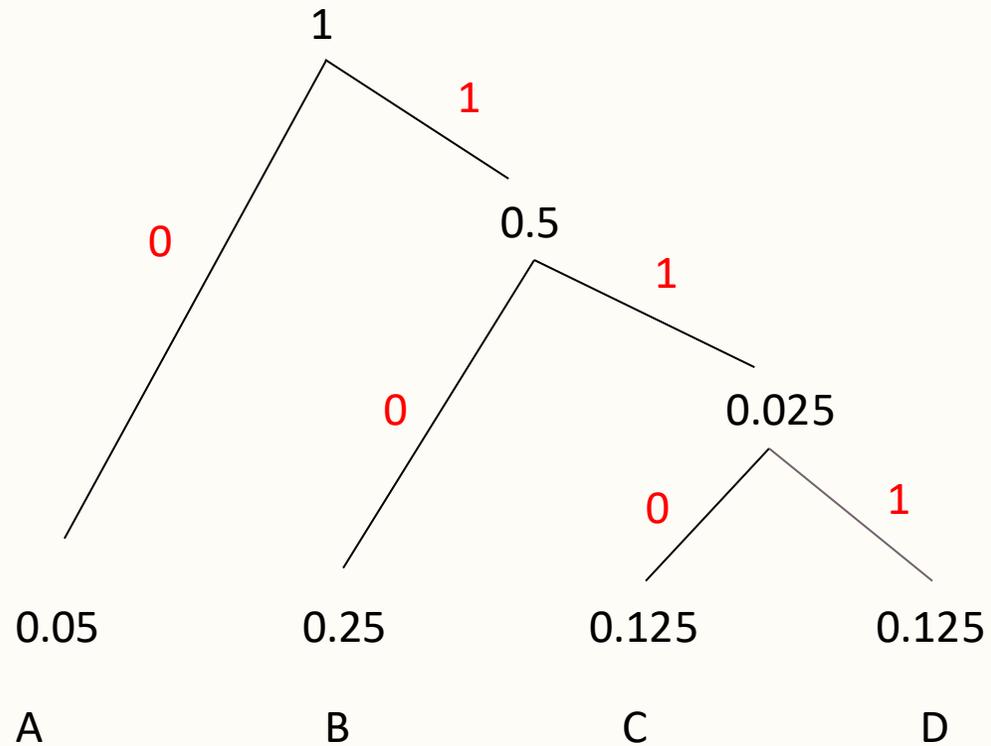
Codage de Huffman

- Exemple décompression : 1101011101000110



Codage de Huffman

- Exemple décompression: 1101011101000110 - **CBDABAAC**
- **$T = 16/64 = 0.25$**
- **$G = 1 - 0.25 = 75\%$**



Codage de Huffman

- Exemple:

symboles	p_i						c_i
A	0.4						
B	0.18						
C	0.1						
D	0.1						
E	0.07						
F	0.06						
G	0.05						
H	0.04						

Codage de Huffman

symboles	p_i							
A	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.6 ⁰
B	0.18	0.18	0.18	0.19	0.23	0.37	0.37 ⁰⁰	0.4 ₁
C	0.1	0.1	0.13	0.18	0.19	0.23	0.23 ⁰⁰⁰	0.23 ₀₁
D	0.1	0.1	0.1	0.13	0.18	0.18	0.18 ⁰¹⁰	0.18 ₀₀₁
E	0.07	0.09	0.1	0.1	0.1	0.1	0.1 ⁰⁰⁰⁰	0.1 ₀₁₁
F	0.06	0.07	0.09	0.09			0.09 ⁰¹⁰⁰	0.09 ₀₀₀₁
G	0.05	0.06					0.06 ⁰⁰⁰¹⁰	0.06 ₀₁₀₁
H	0.04						0.04 ⁰⁰⁰¹¹	

Codage de Huffman

- Exemple:

- ✓ La longueur moyenne de code:

$L_m = 2.61$ bits/symbole

- ✓ L'entropie: $H = 2.55$ bits/symbole

- ✓ L'efficacité du codage est $E = 2.55 / 2.61 \approx 97.8\%$

symboles	p_i	c_i
A	0.4	1
B	0.18	001
C	0.1	011
D	0.1	0000
E	0.07	0100
F	0.06	0101
G	0.05	00010
H	0.04	00011

Codage de Huffman

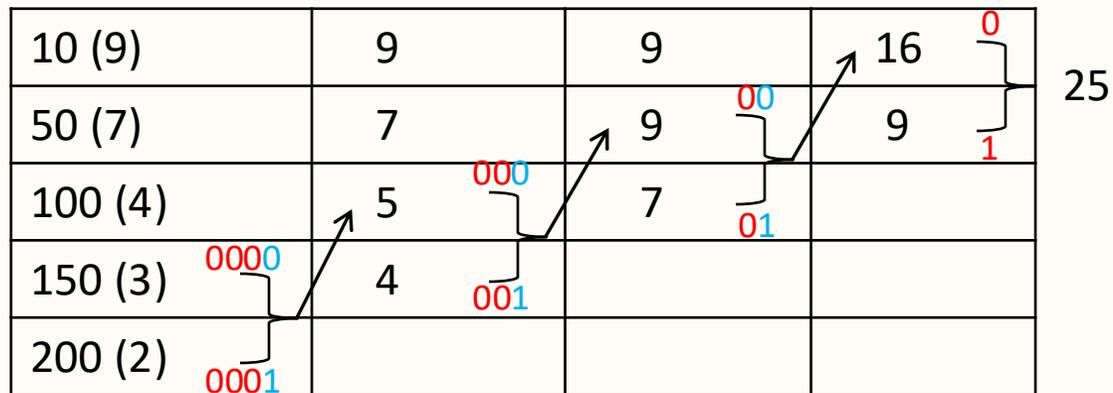
- Exemple: compression d'image

10 (9)			
50 (7)			
100 (4)			
150 (3)			
200 (2)			

10	50	100	100	100
10	50	150	150	50
10	50	150	200	50
10	50	100	200	50
10	10	10	10	10

Codage de Huffman

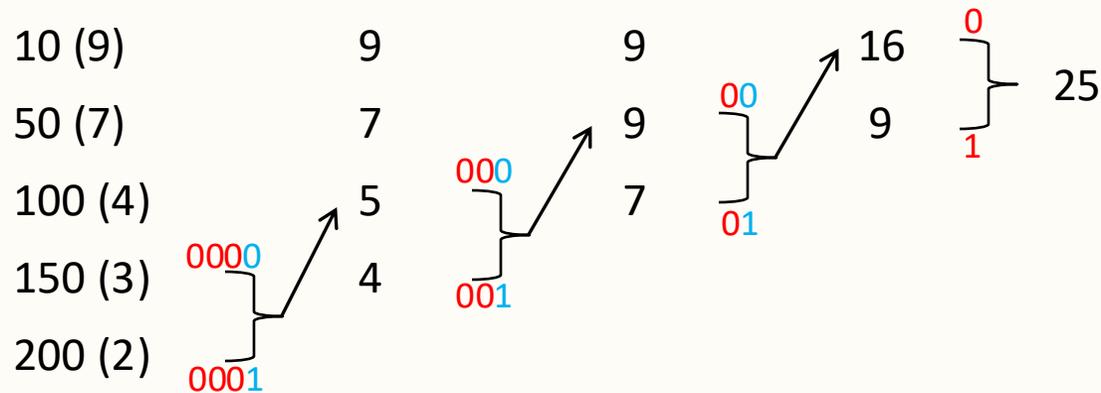
- Exemple: compression d'image



10	50	100	100	100
10	50	150	150	50
10	50	150	200	50
10	50	100	200	50
10	10	10	10	10

Codage de Huffman

- Exemple: compression d'image



Code
1
01
001
0000
0001

10	50	100	100	100
10	50	150	150	50
10	50	150	200	50
10	50	100	200	50
10	10	10	10	10

Image compressée: 55 bits = $9 \cdot 1 + 7 \cdot 2 + 4 \cdot 3 + 3 \cdot 4 + 2 \cdot 4$

Image avants compression: $5 \cdot 5 \cdot 8 = 200$ bits

$$T = 55/200 = 0.275$$

$$G = 1 - T = 72.5\%$$

Codage de Huffman

✓ Avantages:

- Algorithme de compression le plus utilisé (ZIP, JPEG, MP3).
- Fournit une compression optimale pour le codage à longueur variable.
- Garantit qu'aucune donnée n'est perdue pendant la compression.
- Adapté à différents types de données, et flexible avec différentes tailles de fichiers.
- Très bon compromis entre temps d'exécution et taux de compression.