

Compression

Guillaume TOCHON

17 mars 2018

Table des matières

1	Compression de données	1
1.1	Sur la compressabilité	1
1.1.1	Entropie	1
1.1.2	Probabilités	2
1.1.3	Application à l'informatique	3
1.1.4	Cas de l'entropie nulle (À corriger avec le prof)	3
1.1.5	Calcul de l'entropie	4
1.1.6	Exercice	4
2	Old	5

1 Compression de données

Site des slides : www.lrde.epita.fr/~gtochon/CODO/

Compression + décompression

Sans compression de données, un film 720p d'une heure ferait presque 400Go.

La naissance naïve de la compression de données remonte aux environs du code morse. Shannon est responsable des fondements mathématiques.

Autres acteurs : - Abraham Lempel - David Huffman - Terry Welch

Le but de la compression est que lors de la décompression, ce soit le moins visible possible par l'utilisateur.

1.1 Sur la compressabilité

1.1.1 Entropie

On va chercher à éliminer les redondances, et les gaspillages. L'outil que l'on va utiliser est l'entropie.

Du point de vue de Shannon, plus un message est probable, moins il contient d'informations.

Prenons un alphabet Σ de N symboles

$$\Sigma = \{s_1, s_2, \dots, s_N\}$$

Avec leurs probabilités d'occurrence respectives

$$p_i = p(s_i)$$

Évidemment, on a

$$\sum_{i=1}^N p_i = 1$$

Soit F un fichier composé de N_F éléments de Σ .

Statistiquement, s_i est présent $p_i \times N_F$ fois.

On définit q_i la quantité d'information totale d'un symbole :

$$q(s_i) = -\log_2(p_i)$$

On a donc Q_{Tot} la quantité d'information propre totale de s_i contenue dans F :

$$Q_{Tot}(s_i) = -N_F \cdot p_i \cdot \log_2(p_i)$$

On définit donc $Q_{Tot}(F)$ la quantité d'information contenue dans un fichier F :

$$\begin{aligned} Q_{Tot}(F) &= \sum_{i=1}^{N_F} (Q_{Tot}(s_i)) \\ &= \sum_{i=1}^{N_F} -N_F \cdot p_i \cdot \log_2(p_i) \\ &= -N_F \sum_{i=1}^{N_F} p_i \cdot \log_2(p_i) \end{aligned}$$

Or ce $-N_F$ devant la somme pose problème : **la quantité d'information dépend de la taille du fichier**. Cela implique qu'un gros fichier "probable" contient moins d'information qu'un petit fichier improbable.

On définit donc l'entropie H ainsi :

$$H = \frac{Q_{Tot}(F)}{N_F} = - \sum_{i=1}^{N_F} p_i \cdot \log_2(p_i)$$

On notera en outre Q_{Tot} avec le symbole Sh . On a bien évidemment :

$$H \equiv Sh$$

On remarquera que l'entropie H est **indépendante** de la taille du fichier.

H ne dépend donc que de l'alphabet considéré Σ et de la distribution des symboles qui compose le fichier F .

1.1.2 Probabilités

Soit X une variable aléatoire telle que :

$$X = \{x_1, x_2, \dots, x_n\}$$

Et

$$p_i = P(X = x_i)$$

On a donc l'espérance de X :

$$E(X) = \sum_{i=1}^n x_i \cdot p_i = \sum_{i=1}^n x_i \cdot P(X = x_i)$$

Et on a donc :

$$E(\varphi(X)) = \sum_{i=1}^n \varphi(x_i) P(X = x_i)$$

On a donc, pour l'entropie :

$$\begin{aligned}
H &= - \sum_{i=1}^{N_F} p_i \cdot \log_2(p_i) \\
&= \sum_{i=1}^{N_F} \underbrace{(-\log_2(p_i))}_{q_i} \cdot p_i \\
&= \sum_{i=1}^{N_F} q_i \cdot p_i \\
&= E(q(s_i))
\end{aligned}$$

1.1.3 Application à l'informatique

On considère donc :

$$\begin{aligned}
N_\Sigma &= \{0, 1\} \\
p(0) &= p_0 = p \\
p(1) &= p_1 = 1 - p
\end{aligned}$$

On a alors :

$$\begin{aligned}
H &= -p \cdot \log_2(p) - (1 - p) \log_2(1 - p) \\
&= H(p)
\end{aligned}$$

Par convention, on pose :

$$\begin{aligned}
H &= - \sum_{i=1}^{N_F} p_i \cdot \log_2(p_i) \\
p_i \cdot \log_2(p_i) &= 0 \quad \text{si} \quad p_i = 0
\end{aligned}$$

1.1.4 Cas de l'entropie nulle (À corriger avec le prof)

$$H(0) = H(1) = 0$$

On a des fichiers complètement désordonnés :

$$p = 0 \implies F = \{1111...1\}$$

$$p = 1 \implies F = \{0000...0\}$$

Donc l'entropie est nulle.

Quand $p = \frac{1}{2}$, les symboles sont équiprobables, le fichier est complètement désordonné, donc complètement aléatoire. Dans ce cas, l'entropie est donc **maximale**.

Voir la figure [1](#).

Attention ! L'entropie ne voit pas les méta-symboles ! Par exemple, ce fichier est considéré comme parfaitement aléatoire :

$$F = \{01010101\}$$

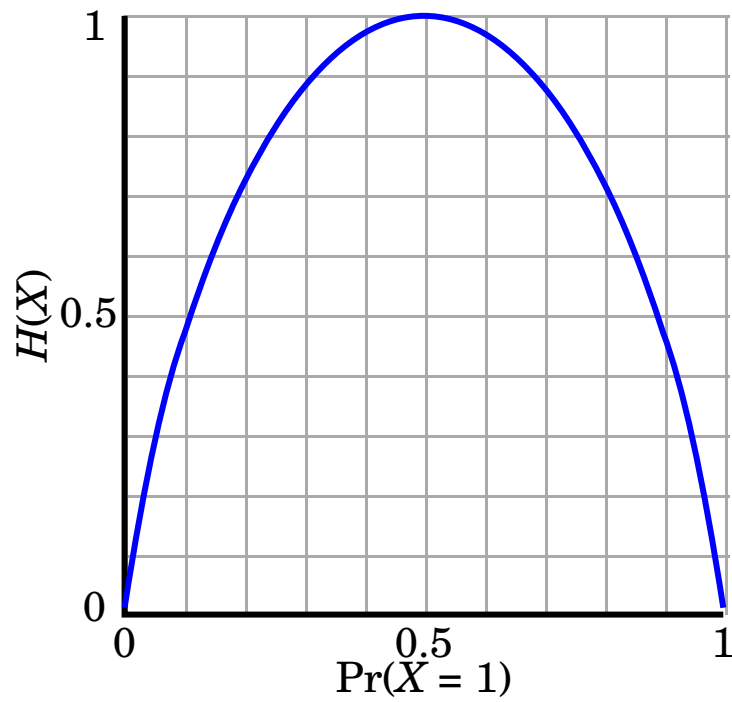


FIGURE 1 – Entropie en fonction de $P(X = 1)$

1.1.5 Calcul de l'entropie

Soit l'alphabet

$$\Sigma = \{s_1, s_2, \dots, s_N\}$$

De longueur :

$$\forall n \in \mathbb{N}, \quad N_\Sigma = 2^n$$

Avec :

$$\forall s_i \in \llbracket 1, N_\Sigma \rrbracket, \quad p(s_i) = p_i = \frac{1}{N_\Sigma}$$

On a donc :

$$\begin{aligned} H &= - \sum_{i=1}^{N_\Sigma} p_i \cdot \log_2(p_i) \\ &= - \sum_{i=1}^{2^n} \frac{1}{2^n} \cdot \underbrace{\log_2\left(\frac{1}{2^n}\right)}_{-n} \\ &= \frac{1}{2^n} (-n) \underbrace{\sum_{i=1}^{2^n} 1}_{2^n} \\ &= n \end{aligned}$$

1.1.6 Exercice

VÉRIFIER AVEC LE PROF !

De combien de bits a-t-on besoin pour encoder le fichier suivant ?

$$F = \{ACABBDDDBAAABC AAA\}$$

Dans ce cas, on utilisera l'alphabet suivant :

$$\Sigma = \{A, B, C, D\}$$

Comptons le nombre d'occurrences de chacune des lettres de Σ :

$$\left. \begin{array}{l} A : 8 \\ B : 4 \\ C : 2 \\ D : 2 \end{array} \right\} \text{Longueur : 16}$$

1. Non compressé ?

C'est tout simplement, en notant Σ l'alphabet utilisé pour encoder le fichier,

$$\begin{aligned} \text{Résultat} &= \text{Longueur} \times \text{Nombre de bits nécessaire pour encoder } \Sigma \\ &= 16 \cdot \log_2(n_\Sigma) \\ &= 16 \cdot \log_2(4) \\ &= 32 \text{ bits} \end{aligned}$$

2. Compressé ?

Calculons l'entropie du fichier :

$$\begin{aligned} H &= p_A \cdot \log_2(p_A) + p_B \cdot \log_2(p_B) + p_C \cdot \log_2(p_C) + p_D \cdot \log_2(p_D) \\ &= \frac{1}{2} \cdot \log_2\left(\frac{1}{2}\right) + \frac{1}{4} \cdot \log_2\left(\frac{1}{4}\right) + \frac{1}{8} \cdot \log_2\left(\frac{1}{8}\right) + \frac{1}{8} \cdot \log_2\left(\frac{1}{8}\right) \\ &= \frac{7}{4} \end{aligned}$$

Donc la longueur totale minimale du message est :

$$\begin{aligned} \text{Résultat} &= \text{Longueur} \times H \\ &= 16 \cdot \frac{7}{4} \\ &= 28 \text{ bits} \end{aligned}$$

2 Old

$$Q_{Tot}(S_i) = -N_{FP_i} Sh$$

$$Q_{Tot}(F) = Q_{Tot}(S_1) + Q_{Tot}(S_2) + \dots + Q_{Tot}(S_{N_2})$$

$$Q_{Tot}(S_1) = -N_F \log_2(P_i)$$

$$Q_{Tot}(S_2) = -N_{FP_2} \log_2(P_2)$$

$$Q_{Tot}(S_{N_2}) = -N_{FP_N} \log_2(P_N)$$

$$Q_{Tot}(F) = \sum -N_{FP_i} = -N_F \sum P_i \log_2(P_i) Sh$$

Un gros fichier "probable" contient plus d'info qu'un petit fichier "improbable" $Q_{Tot} \rightarrow Sh$

$\rightarrow H \rightarrow \text{Sh/Symbole}$ $H = -\sum P_i \log_2(P_i) \rightarrow$ ne depend pas du fichier considere, mais uniquement de la distribution de proba des symboles composant le fichier.

X variable aleatoire de valeur $\{x_1, x_2 \dots x_n \rightarrow P_i = P(X = x) \sum P_i = 1$

$$E[X] = \sum_{i=1}^N x_i P(X = x_i) = \sum_{i=1}^N x_i P_i \quad E[\phi(X)] = \sum_{i=1}^N \phi(x_i) P_i \quad H = -\sum_{i=1}^N P_i \log_2 P_i = \sum_{i=1}^N (-\log_2(P_i)) P_i = \sum_{i=1}^N q_{S_i} * P_i \quad H = E[q(s_i)] \quad \sum = 0, 1 \quad P(0) = P_0 = P \quad P(1) = P_1 = 1 - P \quad H = -p \log_2(p) - (1-p) \log_2(1-p) = H(p)$$

— Missing things —

\sum avec N symboles $s_i, i = 1, \dots, N \quad (2^n)$

$$P(S_i) = \frac{1}{N \sum} = \frac{1}{2^n}$$

$$H = -\sum_{i=1}^{N \sum} \log_2(P_i) = -\sum_{i=1}^{2^n} = -\sum_{i=1}^{2^n} \frac{1}{2^n} \log_2\left(\frac{1}{2^n}\right) = \frac{1}{2^n} * (-n) * \sum_{i=1}^{2^n} * \sum_{i=1}^{2^n} 1 = n 2^n 2^{-n} = n$$

(Sh/Symbole)