COMPRESSION DE HUFFMAN

Rachid Bouyekhf
GI-UV : IN42 2001

Introduction

Depuis ses débuts, l'informatique est confrontée au problème du stockage et de la transmissions des données d'une façon efficace et économique, d'autant plus que la taille des informations ne cesse d'augmenter avec le temps. Le problème était particulièrement flagrant à l'époque où les mémoires de masses (disques durs, bandes magnétiques...) faisaient difficilement quelques Mo, la compression de données s'imposa comme une solution très économique et simple d'utilisation. Aujourd'hui encore, bien que la taille des unités de stockage ait connu un véritable boom, et les vitesses de transmission toujours plus grande, elle présente un très grand intérêt, en permettant par exemple de réduire considérablement le temps de téléchargement à partir d'Internet, ou en simplifiant l'archivage par réduction du nombre de supports nécessaires.

La compression est basée sur une constatation simple: la présence de répétitions fréquente (comme celle des lettres e, a, ... dans un texte français) ou d'une certaine logique dans les documents.

On peut distinguer les compressions conservatrices, restituant l'intégralité du document original après décompression, et les compressions non conservatrices, qui perdent une partie des informations de façon peu visible (comme le jpeg), profitant des défauts de nos sens (par exemple deux points de petite taille, très proches et de couleurs différentes ne semblent en faire qu'un si l'on ne se trouve pas assez près : c'est un principe utilisé par la télévision couleur…).

L'efficacité de la compression est mesurée avec le taux de compression :

$$TC = 100 \times \left(1 - \frac{TailleCompressée}{TailleOriginelle}\right)$$

Remarque: Dans l'explication des compressions suivante, l'octet sera utilisé comme donnée de base, mais les méthodes peuvent être utilisées avec des mots binaires
de longueur quelconque.

La méthode de HUFFMAN

a) Principe

Cette compression à été inventée par David HUFFMAN en 1952, à partir de la constatation de l'inégalité des fréquence d'apparition des données (i.e. les valeurs des octets), au sein d'un même document (par exemple les lettres dans un texte), bien que ces données aient la même taille:
L'idée est de coder les valeurs apparaissant souvent avec moins de bits que celles apparaissant rarement, suivant la règle: "à fréquence élevée, code court". Ainsi une valeur fréquente peut être codée avec moins que 8 bits et une autre très peu fréquente être codée sur plus de 100 bits

L'algorithme de compression est le suivant:

1- établir une table des fréquence d'apparition des valeurs dans le document
2- construire l'arbre de Huffman à partir de cette table, c'est un arbre pondéré où chaque feuille correspond à une valeur, et est un nœud pondérée de la fréquence de cette valeur, les nœuds pères sont obtenus en reliant les deux nœuds inférieurs de pondération la plus faible et en associant au nœud ainsi formé une pondération égale à la somme des pondérations des deux nœuds fils.
3- Il suffit ensuite de parcourir l'arbre de la racine vers la feuille correspondant à la valeur à coder, chaque nœud correspondant à un bit, dont la valeur dépend de la direction prise. (dans la pratique, on part d'une feuille et on parcourt l'arbre jusqu'à la racine, puis on inverse le mot binaire ainsi obtenu)

L'algorithme de décompression est très simple:

1- reconstruction de l'arbre de Huffman (à partir d'information contenues dans l'entête du fichier)
2- lecture séquentielle des bits des données compressées, et parcourt simultané de l'arbre, la direction prise à chaque nœud étant fonction de la valeur du bit, jusqu'à une feuille, comportant la valeur décompressée.

b) Exemple de compression par la méthode de Huffman

Soit les caractères: A B C D

Exemple de Table de correspondances:
(à chaque caractère correspond un nombre binaire) 4 = 2^2 possibilités, donc les lettres peuvent être codés sur 2 Bits.

A <=> 0 = 00
B <=> 1 = 01
C <=> 2 = 10
D <=> 3 = 11

Soit la chaîne:

AABCAABADBACAAB

Codage normal :

00 00 01 10 00 00 01 00 11 01 00 10 00 00 01

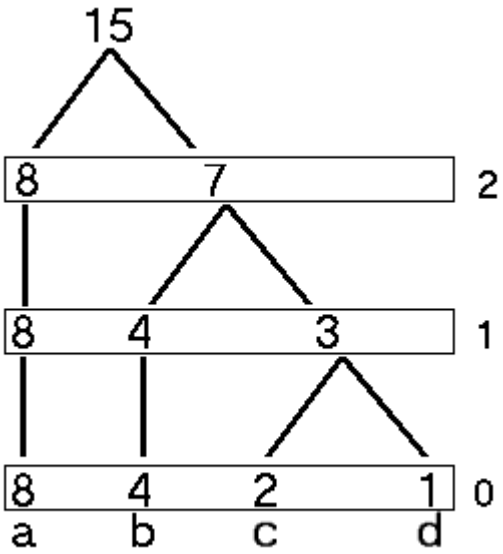
soit 2*15=30 bits

Compression Huffman :

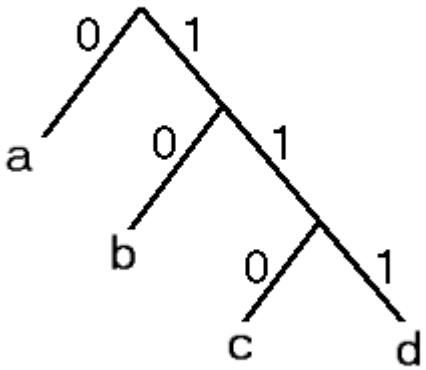
Table des fréquences :

A : 8
B : 4
C : 2
D : 1

Construction de l'arbre binaire :
- Les Feuilles sont constituées des caractères, pondérés de leur fréquence;
- Les Feuilles ou nœuds de poids le plus faible sont regroupés en un nœud de poids égal à la somme des poids:



- Aux branches situées à gauche on associe les Bits de valeur 0, à celles situées à droite, les Bits de valeur 1:



La table d'équivalence est alors (il suffit de parcourir l'arbre, de la racine vers la feuille souhaitée):

A <> 0
B <> 10
C <> 110
D <> 111

(la règle: "à fréquence élevée, code court" est respectée)

La chaîne:
AABCAABADBACAAB
devient alors:

0 0 10 110 0 0 10 0 111 10 0 110 0 0 10

soit 25 Bits

c'est à dire une compression de (30-25)/30=16.6% , soit un taux de compression de 30/25=1.2