

TIPE : Compression de données de type *texte* par méthode adaptative sans perte utilisables en temps réel

Thomas BAGREL,
Lycée H. POINCARÉ, NANCY

Année 2017 - 2018

Contents

| | | |
|----------|--------------------------------------|----------|
| I | Introduction et prérequis | 5 |
| 1 | État de l'Art | 7 |
| 1.1 | Codage de Huffman | 7 |
| 1.1.1 | Principe de fonctionnement | 7 |
| 1.1.2 | Différentes versions | 7 |
| 1.2 | Codage arithmétique | 8 |
| 1.3 | TODO | 8 |

Part I

Introduction et prérequis

Chapter 1

État de l'Art

1.1 Codage de Huffman

1.1.1 Principe de fonctionnement

Le codage de Huffman repose sur un principe simple : coder les caractères (ou les séquences de caractères) par un mot-code dont la longueur est directement liée à la fréquence dudit objet.

“Objet” désignera ici un caractère ou une séquence de caractères, en fonction du mode choisi.

1.1.2 Différentes versions

Huffman statique

Ici, les “fréquences” des différents objets sont *a priori* connues avant la lecture de la source. Le code de Huffman correspondant est donc établi avant le début de la compression. Il est nécessaire que l'algorithme de décodage possède également lesdites fréquences. En revanche, cet algorithme permet de travailler sur un flux de données en temps réel, puisque aucune pré-lecture n'est nécessaire.

Cet algorithme est donc intéressant en première approche, mais ne peut être considéré dans cet exposé puisque ne pouvant se révéler efficace pour des données *texte* quelconques.

Huffman semi-adaptatif

Ici, les fréquences des différents objets sont d'abord calculées lors d'une première lecture de la source, puis le code de Huffman alors calculé est utilisé pour compresser le message. Cependant, il sera nécessaire de transmettre, avec le message compressé, le jeu de fréquences calculées ou le code de message correspondant pour la décompression.

Cet algorithme ne peut donc pas travailler sur un flux de données en temps réel, puisqu'une lecture complète de la source est nécessaire, et ne sera donc pas étudié ici.

Huffman adaptatif

Ici, les fréquences des objets et donc le code de Huffman correspondant sont mis à jour au fur et à mesure de la lecture de la source. Le message compressé obtenu ne nécessite ni la connaissance d'un jeu de fréquences ni une information quelconque sur le type de message transmis pour être efficace. En contrepartie, le processus de mise à jour de l'arbre d'Huffman est plutôt lourd. C'est alors, selon Wikipédia, le processus de compression de type Huffman qui permet d'obtenir les meilleurs résultats.

Cet algorithme rentre donc dans le cadre de notre étude. Nous redétaillerons le principe de fonctionnement exact dans la partie relative à l'étude précise et l'implémentation d'un tel algorithme.

1.2 Codage arithmétique

Le codage arithmétique repose sur le codage des objets sous formes de flottants portant chacun jusqu'à une dizaine d'objets (à cause des contraintes physiques de stockage des flottants en mémoire). Il permet d'améliorer les gains obtenus avec Huffman en autorisant le stockage d'objets sur moins que 1 bit, là où Huffman nécessite au minimum 1 bit par objet (longueur minimale possible d'un code significatif par méthode d'Huffman).

Comme pour Huffman, le codage arithmétique peut être décliné en version statique ou semi-statique, qui ne nous intéresse pas ici car nécessitant des connaissances sur le contenu du fichier ou bien une pré-lecture de la source, et en version dynamique, qui reprend le même principe que le codage Huffman dynamique : les probabilités sont mises à jour au fur et à mesure de la lecture et le code de chaque objet évolue donc de ce fait au cours du processus.

Nous redétaillerons le principe de fonctionnement exact dans la partie relative à l'étude précise et l'implémentation d'un tel algorithme.

file:///home/thomas/Bureau/25945.pdf page 22 arret page 24

1.3 TODO

- Explications détaillées de Huffman adaptatif
- Explications détaillées du codage arithmétique adaptatif