Huffman

Auteurs : Quentin Januel, Loïc Mohin et Anthony Villeneuve
Mentors : Olivier Gipouloux et Stéphane Gaussent

Fait à : Université Jean Monnet, Saint Etienne

5 février 2019

Résumé

Dans ce rapport, on se propose d'étudier le codage d'un texte en binaire par les arbres de Huffman. Nous verrons pourquoi ce codage est optimal...

Table des matières

1 Prérequis

Dans cette section, nous allons tâcher de définir les outils dont nous aurons besoin pour l'analyse des arbres de Huffman.

1.1 Alphabet

On appelle Σ un alphabet dont les éléments sont appelés des lettres.
Un mot sur Σ est un n-uplet de lettres : $m = (a_1,\ a_2,\ ...,\ a_n)$.
L'ensemble des mots sur Σ est noté $\Sigma^* := \{m \in \Sigma^n,\ \forall n \in \mathbb{N}\}$.
Soit $m = (a_1,\ a_2,\ ...,\ a_n)$ un mot, on appelle longueur du mot m notée $|m|$ l'entier n.
Enfin, on note ε le mot vide (unique mot de longueur 0).

On peut munir Σ^* d'une loi de composition interne, la concaténation $+$:
$(a_1,\ ...,\ a_n) + (b_1,\ ...,\ b_n) = (a_1,\ ...,\ a_n,\ b_1,\ ...,\ b_n)$.
On observe alors que $(\Sigma^*,\ +)$ est un monoïde.

1.2 Mot pondéré

On dit qu'un élement $(a,\ n) \in \Sigma^* \times \mathbb{N}$ est un mot pondéré de Σ et on notera :

1. $l((a,\ n)) = a$ (l pour "lettre"),
2. $p((a,\ n)) = n$ (p pour "poids")

On définit alors la somme de mots pondérés $x+y = (l(x)+l(y),\ p(x)+p(y))$ et on se retrouve avec un nouveau monoïde : $(\Sigma^* \times \mathbb{N},\ +)$.

1.3 Arbre binaire

Soit E un ensemble, on dit que $A := (Q,\ T)$ est un arbre binaire sur E avec $Q \subset E$ et $T \subset E \times \mathbb{F}_2 \times E$ s'il respecte les 3 propriétés suivantes :

1. $\exists\ !\ r \in Q,\ \forall (x,\ b) \in Q \times \mathbb{F}_2,\ (x, b, r) \notin T$ (r est appelée racine de A notée $r(A)$),
2. $\forall x_2 \in Q \backslash \{r\},\ \exists\ !\ (x_1,\ b) \in Q \times \mathbb{F}_2,\ (x_1,\ b,\ x_2) \in T$, (on dit que x_1 est un parent de x_2 que l'on note $\pi(x_2)$),
3. $\forall (x_1,\ b) \in Q \times \mathbb{F}_2,\ \text{card}(\{x_2 \in Q,\ (x_1,\ b,\ x_2) \in T\}) \leq 1$.

Les éléments de Q (notés $q(A)$) sont appelés les états et les éléments de T (notés $t(A)$) transitions.
Enfin, l'ensemble des arbres binaires sur E est noté $\mathcal{A}_E$.

2 Arbre de Huffman

Si l'on prend un texte quelconque et que l'on compte le nombre d'occurences de chaque lettre afin de se retrouver avec une liste de lettres pondérées, considérées comme des arbres binaires à un état, on peut alors construire l'arbre de Huffman de ce texte en fusionnant à chaque étape les 2 arbres dont les poids des racines sont minimum jusqu'à ne se retrouver qu'avec un seul arbre.
Une fusion consiste à rajouter une racine tel que son fils gauche soit l'un des deux arbres et son fils gauche l'autre arbre.

On peut ensuite encoder le texte en suivant le chemin depuis la racine jusqu'à chaque lettre en ajoutant un 0 par transition à gauche et un 1 pour la droite.
Un tel arbre n'est pas unique car on peut choisir les arbres à poids minima s'il y en a plusieurs, de plus il n'y a pas de restriction sur quel arbre mettre à gauche ou à droite au moment de fusionner.
L'objectif sera donc de montrer que ces choix n'affectent pas le nombre de bits nécessaires pour l'encodage.

2.1 Définition

Tâchons d'abord de définir quels arbres parmi les arbres binaires sont des arbres de Huffman.
Prennons un alphabet Σ quelconque. Tout arbre de la forme

$$(\{x\},\ \emptyset),\ x \in \Sigma^* \times \mathbb{N},\ |x| = 1$$

sera appelé arbre de Huffman sur Σ.
De plus, soient A et B deux arbres de Huffman et $r := r(A) + r(B)$, alors

$$M_{A,\ B} := (q(A) \cup q(B) \cup \{r\},\ t(A) \cup (B) \cup \{(r,\ 0,\ r(A)),\ (r,\ 1,\ r(B))\})$$

est également un arbre de Huffman et on dit que M est la fusion de A et de B.
Notons $\mathcal{H}_\Sigma$ l'ensemble des arbres de Huffman sur Σ.
On pose aussi

$$\begin{array}{rccc} m: & \mathcal{H}_\Sigma \times \mathcal{H}_\Sigma & \to & \mathcal{H}_\Sigma \\ & (A,\ B) & \mapsto & M_{A,\ B} \end{array}$$

2.2 Propriété

Pour tout alphabet Σ, on a $\mathcal{H}_\Sigma \subset \mathcal{A}_{\Sigma^* \times \mathbb{N}}$.
Preuve :

A faire (facile, montrer que $M_{A,\ B}$ respecte les 3 propriétés d'un arbre binaire)

2.3 Classification des arbres de Huffman

L'objectif est de construire des classes d'équivalence d'arbres de Huffman selon le nombre de bits qu'ils encodent.
L'ensemble des feuilles d'un arbre binaire A est

$$\mathcal{F}_A := \{x_1 \in q(A),\ \forall (x_1,\ b) \in q(A) \times \mathbb{F}_2,\ (x_1,\ b,\ x_2) \notin t(A)\}$$

Soit ω une fonction qui à un arbre lui associe cette longueur d'encodage, on a par définition

$$\begin{array}{rccc} \omega : & \mathcal{H}_\Sigma & \to & \mathbb{N} \\ & A & \mapsto & \displaystyle\sum_{x \in \mathcal{F}_A} n_x \times p(x) \end{array}$$

où n_x est la profondeur de x, c'est-à-dire l'entier n tel que $\pi^n(x) = r(A)$.
On définit la relation d'équivalence

$$A \mathcal{R} B \iff \omega(A) = \omega(B),\ \forall A,\ B \in \mathcal{H}_\Sigma$$

On dénote également $\overline{\mathcal{H}_\Sigma} := \mathcal{H}_\Sigma / \mathcal{R}$ l'ensemble quotient de $\mathcal{H}_\Sigma$ par $\mathcal{R}$.

2.4 Lemme 1

$$\forall A \in \mathcal{H}_\Sigma,\ \omega(A) = \sum_{(x_1,\ b,\ x_2) \in t(A)} p(x_2)$$

Preuve :
A faire

2.5 Lemme 2

$$\forall A,\ B \in \mathcal{H}_\Sigma,\ \omega \circ m(A,\ B) = \omega(A) + p \circ r(A) + \omega(B) + p \circ r(B)$$

Preuve :
A faire

2.6 Théorème

Pour tout mot de Σ, l'arbre de Huffman associé est unique dans $\overline{\mathcal{H}_\Sigma}$.

Preuve :
A faire