Les données sont D[1]

Étude de méthodes de compression sans pertes

Thomas Bagrel

Lycée Henri Poincaré, Nancy

TIPE session 2018

Aperçu

Régularités et gains
- Théorie
- `zip` recursif

Composantes de la compression

Codage
- Problème résolu
- Inefficacité de HUFFMAN – pourquoi

Modèles généraux
- `bitwise encoder` et `ppm`
- `bitwise ppm` et `bitwise ppm flat`

Impact de la transformée
- `bwt` régularise nos données
- En action avec la `rle`

Résultats finaux

Régularités et gains

Théorie

Compression des données sans pertes

Régularités et gains

Théorie

Compression des données sans pertes

- exploiter les régularités des données

Régularités et gains

Théorie

Compression des données sans pertes

- exploiter les régularités des données
- données aléatoires : pas de gain

Régularités et gains

Théorie

Compression des données sans pertes

- exploiter les régularités des données
- données aléatoires : pas de gain

Théorème. (Entropie de SHANNON)

$$\mathrm{H}(\mathrm{S}) = -\sum_{i=1}^{n} p_i \log_2(p_i)$$

$\mathrm{H}(\mathrm{S})$: *nb. de bits moyen par symbole de la source*

Régularités et gains

Théorie

Compression des données sans pertes

- exploiter les régularités des données
- données aléatoires : pas de gain

Théorème. (Entropie de Shannon)

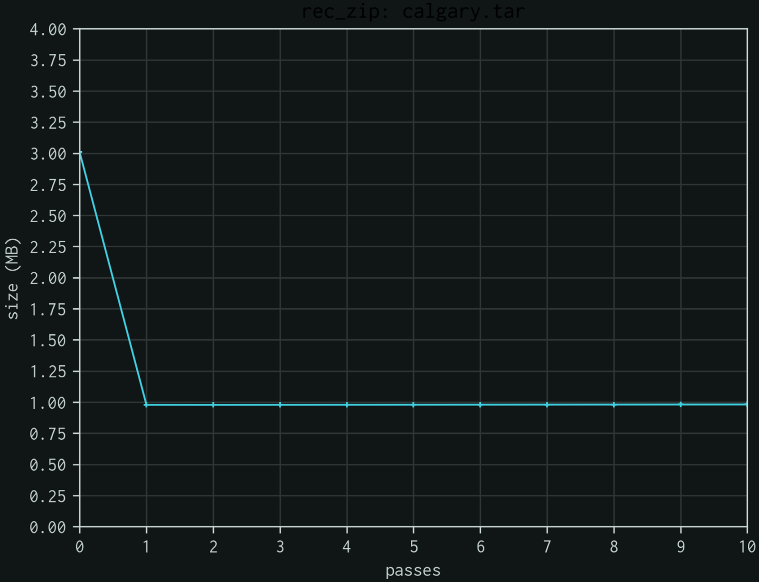
$$\mathrm{H}(\mathrm{S}) = -\sum_{i=1}^{n} p_i \log_2(p_i)$$

$\mathrm{H}(\mathrm{S})$: *nb. de bits moyen par symbole de la source*

- une fois les données compressées (dérivées) une fois, plus aucune régularité

Régularités et gains

zip recursif



Composantes de la compression

(Transformée)

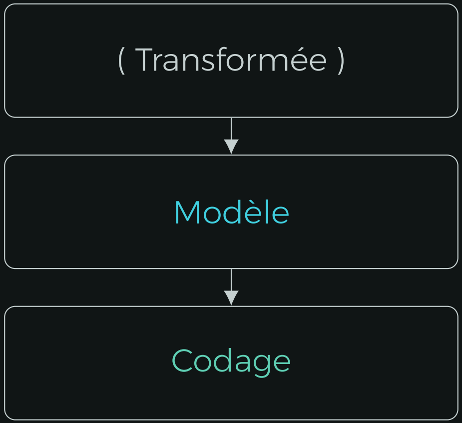
Composantes de la compression

(Transformée)

↓

Modèle

Composantes de la compression



Codage

Problème résolu

Le codage, contrairement aux apparences, est un problème résolu

Codage

Problème résolu

Le codage, contrairement aux apparences, est un problème résolu

- Si les p_i sont connus, la limite de compression théorique est donnée par Shannon

Codage

Problème résolu

Le codage, contrairement aux apparences, est un problème résolu

- Si les p_i sont connus, la limite de compression théorique est donnée par SHANNON
- HUFFMAN permet d'approcher cette limite

Codage

Problème résolu

Le codage, contrairement aux apparences, est un problème résolu

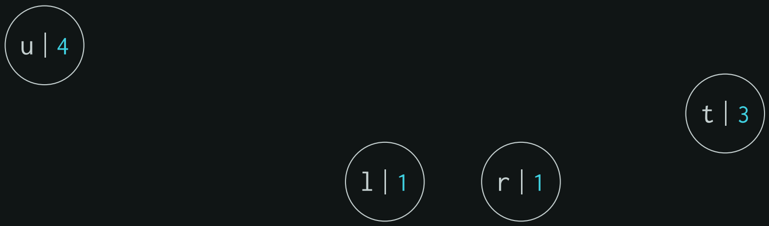
- Si les p_i sont connus, la limite de compression théorique est donnée par SHANNON
- HUFFMAN permet d'approcher cette limite
- Le codage arithmétique l'atteint

Codage

Problème résolu

Le codage arithmétique sera détaillé plus loin.

Codage d'HUFFMAN pour `turlututu`

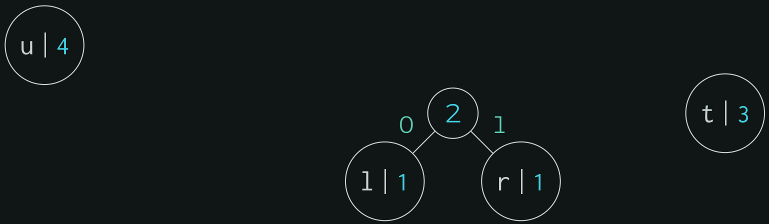


Codage

Problème résolu

Le codage arithmétique sera détaillé plus loin.

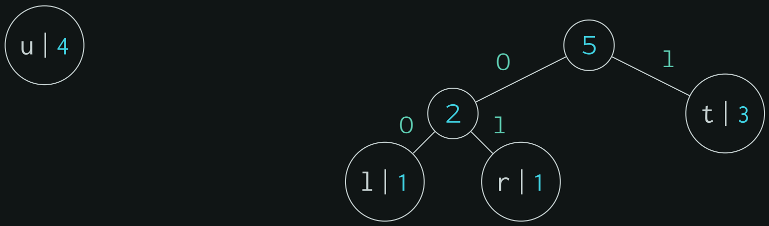
Codage d'Huffman pour `turlututu`



Codage

Problème résolu

Le codage arithmétique sera détaillé plus loin.
Codage d'HUFFMAN pour `turlututu`

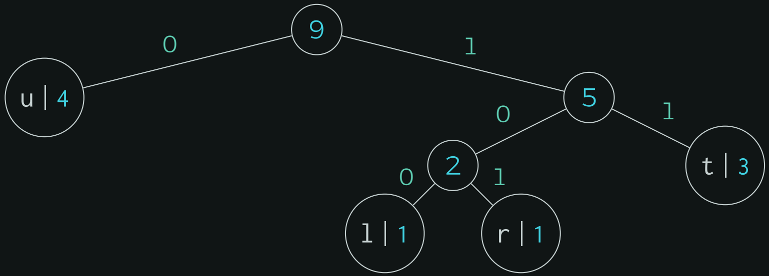


Codage

Problème résolu

Le codage arithmétique sera détaillé plus loin.

Codage d'HUFFMAN pour `turlututu`

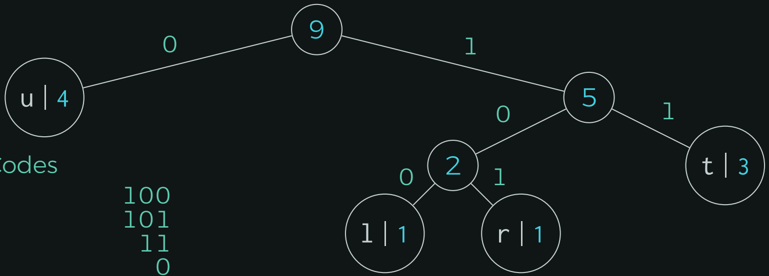


Codage

Problème résolu

Le codage arithmétique sera détaillé plus loin.

Codage d'HUFFMAN pour `turlututu`



Codes

l	100
r	101
t	11
u	0

Codage

Inefficacité de Huffman – pourquoi

En pratique, efficacité de 32 %

Codage

Inefficacité de Huffman – pourquoi

En pratique, efficacité de 32 %

En appliquant directement Shannon aux fréquences d'apparition, on commet des erreurs

Codage

Inefficacité de Huffman – pourquoi

En pratique, efficacité de 32 %

En appliquant directement Shannon aux fréquences d'apparition, on commet des erreurs

- un symbole n'est pas indépendant des précédents

Codage

Inefficacité de HUFFMAN – pourquoi

En pratique, efficacité de 32 %

En appliquant directement SHANNON aux fréquences d'apparition, on commet des erreurs

- un symbole n'est pas indépendant des précédents
- par exemple, en Français, q→u est plus fréquent que q→z

Codage

Inefficacité de HUFFMAN – pourquoi

En pratique, efficacité de 32 %

En appliquant directement SHANNON aux fréquences d'apparition, on commet des erreurs

- un symbole n'est pas indépendant des précédents
- par exemple, en Français, q→u est plus fréquent que q→z
- en quelque sorte, on oublie le caractère lipschitzien de nos données

Codage

Inefficacité de HUFFMAN – pourquoi

En pratique, efficacité de 32 %

En appliquant directement SHANNON aux fréquences d'apparition, on commet des erreurs

- un symbole n'est pas indépendant des précédents
- par exemple, en Français, q→u est plus fréquent que q→z
- en quelque sorte, on oublie le caractère lipschitzien de nos données

Il faut donc un modèle

Résultats finaux

bitwise ppm flat