Codage de source
MA331

Nicolas Barbot

nicolas.barbot@esisar.grenoble-inp.fr

2021-2022

Compression de données

Compression sans perte

L'information peut être retrouvée en intégralité à partir de la séquence codée

- Format de fichier: .bz2 (TBW+Huffman), .zip, .jar (deflate), rar (LZ+PPM), 7z (LZMA) .X (LZW).

Compression avec perte

L'information ne peut pas être retrouvée en intégralité (cependant le taux de compression est plus important).

- Audio: .mp3 (MPEG-1/2 Audio Layer 3), ogg, wma.
- Image: .jpeg (JPEG, JPEG 2000).
- Video: MPEG 2, MPEG 4, H.264, HEVC.

Codage par symbole

Code symbole binaire

Un code C d'un ensemble X est un mapping entre les symboles de la source X vers une représentation binaire. On appelle $c(x)$ le mot de code correspondant au symbole x et $l(x)$ sa longueur.

Un code étendu est un mapping entre une séquence de symboles vers une représentation binaire obtenue par concaténation des mots de codes de chaque symbole:

$$c^+(x_1x_2x_3\dots x_N) = c(x_1)c(x_2)c(x_3)\dots c(x_N)$$

Code uniquement décodable

Code uniquement décodable

Un code est décodable de manière unique si toutes les séquences de symboles admettent une représentation binaire différente:

$$\forall \mathbf{x}, \mathbf{y} \in \mathcal{A}_X^+, \mathbf{x} \neq \mathbf{y} \Rightarrow c^+(\mathbf{x}) \neq c^+(\mathbf{y})$$

Un code non uniquement décodable est inutilisable en pratique puisqu'il implique une ambiguïté lors du décodage.

Code préfixe

Un code symbole est plus facile à décoder si l'on peut identifier un symbole dès que le mot de code a été reçu.

Code préfixe

Un code est un code préfixe si aucun mot de code n'est le préfixe d'un autre

Un code préfixe peut être décrit par un arbre où tous les mots de codes sont placés sur des feuilles

Longueur moyenne

Longueur moyenne

La longueur moyenne $L(C, X)$ d'un code C pour un ensemble X est définie par:

$$L(C, X) = \sum_{x \in \mathcal{A}_X} p(x) l(x) \tag{1}$$

Cette longueur représente le nombre de bit par symbole moyen nécessaire pour encoder les symboles émis par la source.

En utilisant le 1er théorème de Shannon, la longueur moyenne d'un code sans perte est bornée par:

$$H(X) \leq L(C, X) \tag{2}$$

Exemple

On considère la source X définie par $\mathcal{A}_X = \{a, b, c, d\}$ et $\mathcal{P}_X = \{1/2, 1/4, 1/8, 1/8\}$ ainsi que les codes suivants:

	C_1	C_2	C_3	C_4	C_5
a	00	0001	0	0	0
b	01	0010	10	1	01
c	10	0100	110	00	011
d	11	1000	111	11	111

Calculez la longueur moyenne de chaque code pour la source X.
Encodez la séquence *abaadcda* pour tout les codes
Décodez la séquence 0101001110 pour tout les codes
En déduire les codes préfixes
En déduire les codes uniquement décodables

Inegalité de Kraft McMillan

Inégalité de Kraft

Pour tous les codes (binaires) uniquement décodables, la longueur des mots de code doit satisfaire :

$$\sum_{i=1}^{|\mathcal{A}_X|} 2^{-l_i} \leq 1 \tag{3}$$

De plus, pour un ensemble de mots de code de longueur donnée respectant l'inégalité de Kraft, il existe un code préfixe uniquement décodable avec ces longueurs.

Propriété

Théorème du codage de source pour les codes symbole

Pour un ensemble X, il existe un code préfixe dont la longueur moyenne satisfait :

$$H(X) \leq L(C, X) \leq H(X) + 1 \tag{4}$$

La longueur moyenne est minimisée et égale à $H(X)$ seulement si la longueur des mots de code vaut:

$$l_i = \log_2(1/p_i)$$

Algorithme de Shannon Fano

L'algorithme a été proposé par Fano sur une idée de Shannon.

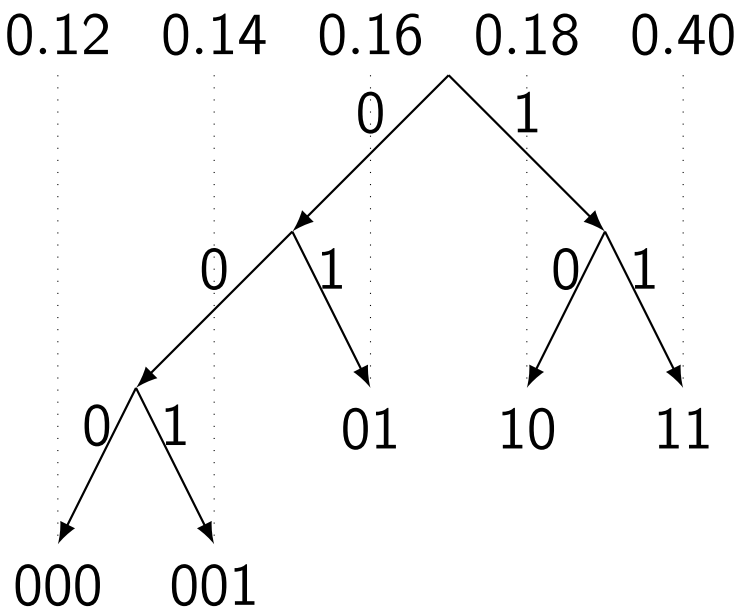
1. Classer les probabilités des message p_i par ordre croissant
2. Diviser la séquence en 2 sous séquences équiprobable
3. Attribuer un label différent à chaque séquence (MSB)
4. Répéter jusqu'a ce que la taille des séquences soit égale à 1

Exemple

On considère la source X définie par $\mathcal{A}_X = \{a, b, c, d, e\}$ et $\mathcal{P}_X = \{0.4, 0.18, 0.16, 0.14, 0.12\}$

Exemple

On considère la source X définie par $\mathcal{A}_X = \{a, b, c, d, e\}$ et $\mathcal{P}_X = \{0.4, 0.18, 0.16, 0.14, 0.12\}$



Propriétés

- Le code produit est un code préfixe (uniquement décodable)
- L'arbre est construit de la racine vers les feuilles
- Les messages plus probables sont codés avec des mots de code plus courts
- L'algorithme n'est pas optimal
- On peut montrer que $L(C, X) \leq H(X) + 2$

Algorithme de Huffman

L'algorithme de Huffman permet de construire le code préfixe optimal d'une source X.
L'algorithme de Huffman est semblable à celui de Shannon Fano à la différence que l'arbre est construit des feuilles vers la racine

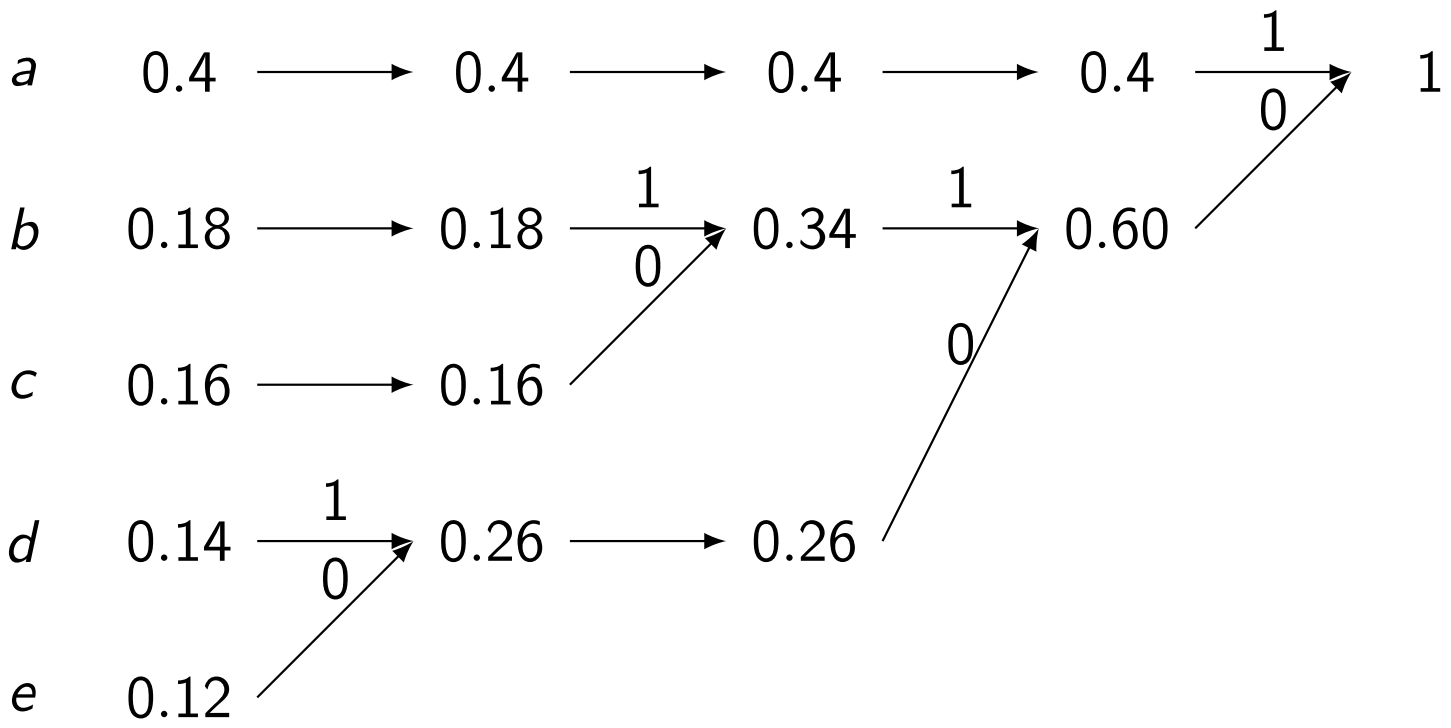
1. Classer les probabilités des messages p_i par ordre croissant
2. Prendre les deux symboles les moins probables et assigner un label
3. Fusionner les deux symboles
4. Répéter tant qu'il y a plus d'un ensemble

Exemple

On considère la source X définie par $\mathcal{A}_X = \{a, b, c, d, e\}$ et $\mathcal{P}_X = \{0.4, 0.18, 0.16, 0.14, 0.12\}$

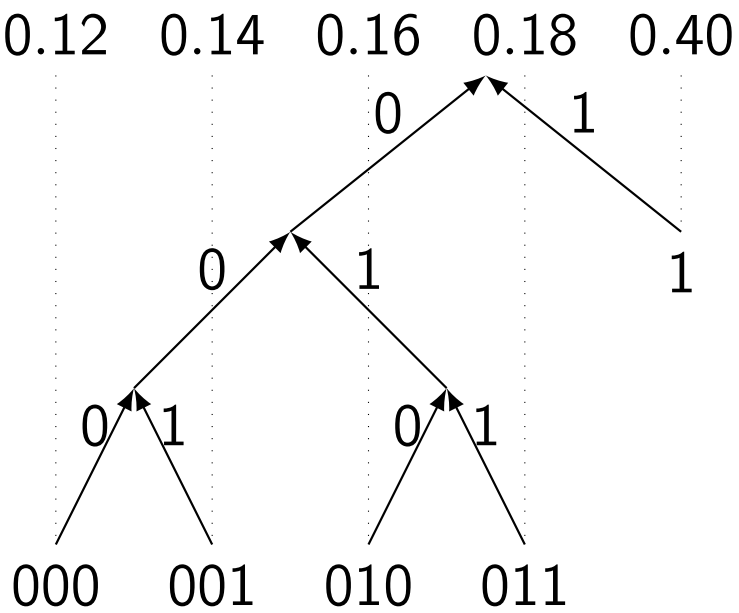
Exemple

On considère la source X définie par $\mathcal{A}_X = \{a, b, c, d, e\}$ et $\mathcal{P}_X = \{0.4, 0.18, 0.16, 0.14, 0.12\}$



Exercices

On considère la source X définie par $\mathcal{A}_X = \{a, b, c, d, e\}$ et $\mathcal{P}_X = \{0.4, 0.18, 0.16, 0.14, 0.12\}$



Calculez $L(C, X)$ pour le codage de Shannon Fano et Huffman
Comparez avec l'entropie de la source $H(X)$

Défauts du codage de Huffman

- Mauvaise adaptation lorsque $\mathcal{P}_X$ n'est pas stationnaire (ou inconnue)
  - Recalculer $\mathcal{P}_X$ lorsque la source change
  - Considérer $\mathcal{P}_X$ moyen sur l'ensemble de la séquence
- La longueur moyenne du code est bornée par $L(C, X) < H(X) + 1$
  - Chaque symbole occupe au moins 1 bit
  - Pénalité pour les sources ayant une entropie faible
  - Possibilité de coder des blocs de symboles

Les défauts du codage de Huffman sont rectifiés par le codage arithmétique

Codage arithmétique

- Inventé par Elias
- Nécessite un modèle prédictif permettant de déterminer les probabilités des messages connaissant la séquence déjà reçue
- Méthode adaptée pour les sources non stationnaires
- L'encodeur crée une séquence binaire à partir du modèle
- Le décodeur retrouve la séquence en utilisant un modèle identique

Un petit jeu

On considère la phrase mystère “BONJOUR A TOUS”. Un individu essaie de la découvrir en devinant les lettres dans l’ordre. On note le nombre de tentatives pour chaque lettre.

B	O	N	J	O	U	R		A		T	O	U	S
10	3	2	1	1	1	1	1	16	1	5	1	1	1

Un petit jeu

On considère la phrase mystère "BONJOUR A TOUS". Un individu essaie de la découvrir en devinant les lettres dans l'ordre. On note le nombre de tentatives pour chaque lettre.

B	O	N	J	O	U	R		A		T	O	U	S
10	3	2	1	1	1	1	1	16	1	5	1	1	1

Le nombre de tentatives semble possèder une entropie plus faible.

On suppose que l'on possède la séquence du nombre de tentatives. Alors, si on utilise le même individu, il est possible de retrouver la phrase mystère en stoppant l'individu lorsque le nombre de tentatives est égale à celui reçu.

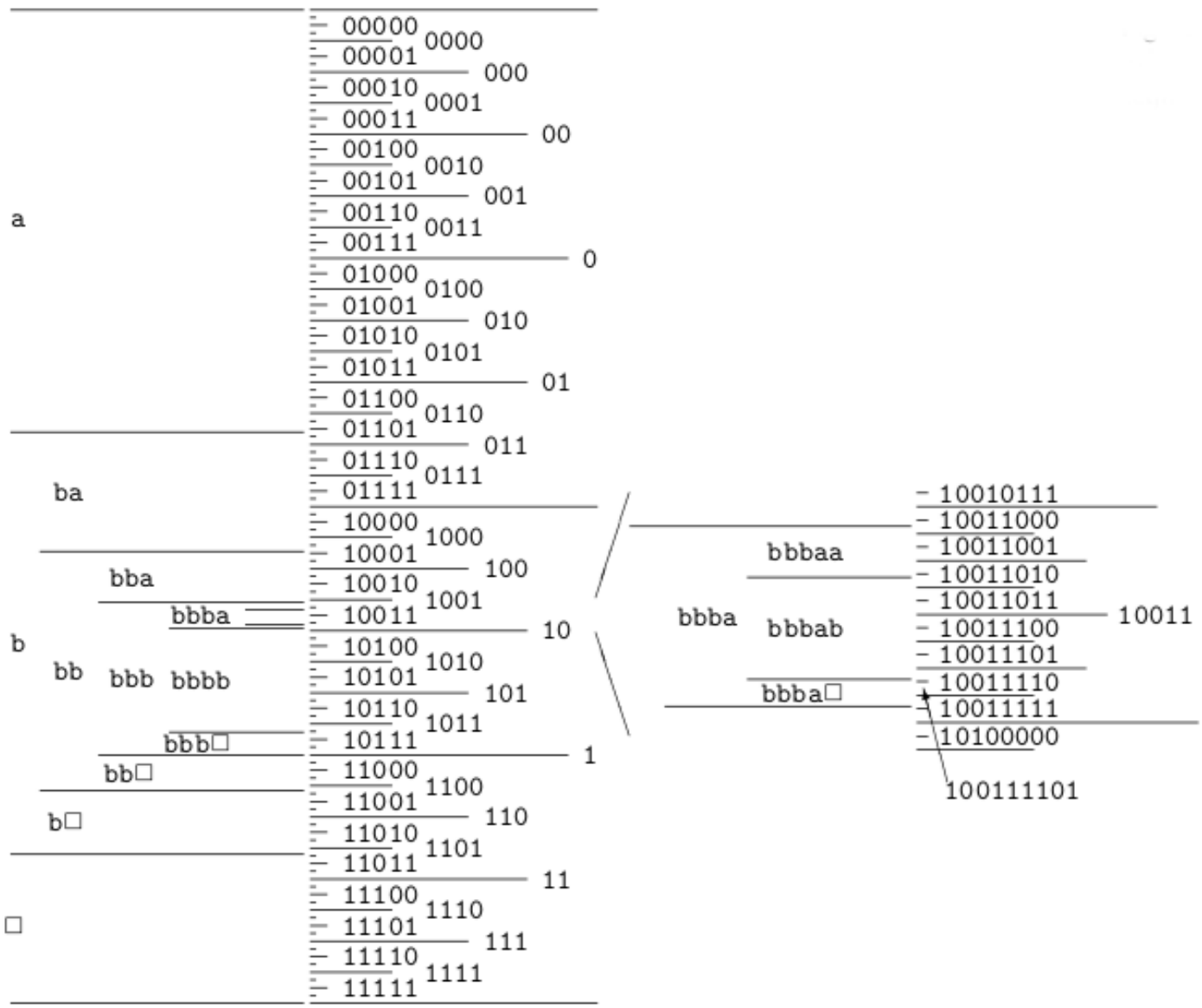
Codage arithmétique

On lance une pièce biaisée 4 fois, les trois sorties possibles sont appelées a, b et $\square$ où $\square$ représente la fin de la série.
La séquence obtenue est $bbba\square$.

Entre chaque lancé, on utilise le modèle pour prédire la probabilité de la sortie suivante. On considère les probabilités suivantes

	$p(a) = 0.425$	$p(b) = 0.425$	$p(\square) = 0.15$
b	$p(a\|b) = 0.28$	$p(b\|b) = 0.57$	$p(\square\|b) = 0.15$
bb	$p(a\|bb) = 0.21$	$p(b\|bb) = 0.64$	$p(\square\|bb) = 0.15$
bbb	$p(a\|bbb) = 0.17$	$p(b\|bbb) = 0.68$	$p(\square\|bbb) = 0.15$
$bbba$	$p(a\|bbba) = 0.28$	$p(b\|bbba) = 0.57$	$p(\square\|bbba) = 0.15$

Codage arithmétique



Modèle

Le modèle utilisé précédemment assigne une probabilité de 0.15 à la sortie □ et pondère le reste pour a et b en utilisant la loi de Laplace :

Modèle bayésien

La probabilité de la sortie a sachant la séquence de symboles déjà reçue vaut :

$$P_L(a|x_1, x_2 \dots x_{n-1}) = \frac{F_a + 1}{F_a + F_b + 2}$$

où F_a et F_b sont le nombre de fois où la sortie a ou b a été observée

Certains codes arithmétiques sont aussi basés sur le modèle de Dirichelet.

Codage de Lempel-Ziv

- Découvert par Lempel et Ziv en 77 et 78
- Codage par flux de données
- Pas de séparation entre le modèle et le codage
- Construction d'un dictionnaire "à la volée" (pour l'encodage et le décodage)
- Algorithme utilisé par `gzip` et `compress`

Encodage

Une source binaire produit la séquence 1011010100010 . . .

L'encodeur crée un dictionnaire contenant les nouvelles sous chaînes: λ, 1, 0, 11, 01, 010, 00, 10, . . . où λ est la sous chaîne vide. Chaque sous chaîne est encodée en envoyant un pointeur plus 1 bit supplémentaire:

Chaine	λ	1	0	11	01	010	00	10
Indice	000	001	010	011	100	101	110	111
pt, bit		0, 1	0, 0	01, 1	10, 1	100, 0	010, 0	001, 0

La séquence codée est donc 010001110110000100010

Décodage

Le décodeur utilise le même algorithme pour recréer une table identique à partir de la chaîne encodée.

Exercices :

- Décodez la chaîne utilisée en exemple
- Encodez et décodez la chaîne 000000000000010000000000

Conclusion

Le codage de source permet de réduire la taille du message envoyé sur le canal.

- Codage de Huffman : sources i.i.d connues
- Codage de Lempel-Ziv : sources non stationnaires ou inconnues

Cependant la séquence codée est extrêmement sensible aux erreurs de transmission (changez la valeur d'un bit sur une séquence codée avec Huffman ou Lempel-Ziv et effectuez le décodage à nouveau pour comparer les deux séquences).

Le codage de canal permet de réduire les erreurs de transmission à une valeur donnée (arbitrairement faible).