Département d'informatique et de génie logiciel

Compression de données
IFT-4003/IFT-7023

Notes de cours
Codes de Golomb-Rice et Codes de Tunstall

Édition Hiver 2012

Mohamed Haj Taieb

Local: PLT 2113

Courriel: mohamed.haj-taieb.1@ulaval.ca

Faculté des sciences et de génie
Département de génie électrique et de génie informatique

UNIVERSITÉ LAVAL

Plan de la présentation

- Codage de Golomb-Rice:
  - Code unaire
  - Code de Golomb
  - Code de Rice
  - Code de Tunstall

Code unaire

- Code de Golomb-Rice
  - Encodage des entiers sous l'hypothèse que les grands entiers ont une probabilité faible.
  - Exemple: encodage de la différence entre les pixels voisins d'une image.
- Code Unaire
  - Le plus simple code pour cette situation est le code unaire.
  - n→11…10 (n fois '1' et un '0' à la fin) i.e. 4→11110
  - Pour l'ensemble semi-infinie {1, 2, 3, …} avec le modèle de probabilité: $P[k]=1/2^k$ (puissance négative de deux), le code unaire coïncide avec le code de Huffman.
  - Comme le code de Huffman est optimal dans cette situation, le code unaire l'est aussi.

Code de Golomb

- Extention des codes unaires
  - Le code est optimal mais uniquement dans une situation très restreinte.
  - Modification du code unaire: partage de l'ensemble des entiers en deux parties.
  - Une partie est codée avec un code unaire.
  - Une seconde partie est codée avec un autre code.
  - Comme exemple d'un tel code on cite le code de Golomb.
- Code de Golomb
  - Code paramétré par un entier strictement positif m>0.
  - Pour un entier n on associe et on encode deux entier q et r tel que:

$$q = \left\lfloor \frac{n}{m} \right\rfloor, \quad r = n - qm$$

Code de Golomb: codage du quotient et du reste

- Codage du quotient

$$q = \left\lfloor \frac{n}{m} \right\rfloor, \quad r = n - qm$$

  - Le quotient q peut prendre des valeurs de {0,1,2,…}.
  - Le quotient q est codé avec un code unaire.
- Codage du reste
  - Le reste r peut prendre m valeurs possibles appartenant à l'ensemble suivant: {0,1,2,…m-1}.
  - Si m est une puissance de 2, on code r avec la représentation binaire de longueur $\log_2 m$.
  - Si m n'est pas une puissance de 2, on peut toujours coder r avec une représentation binaire mais de longueur $\lceil \log_2 m \rceil$
  - Exemple si m=5→ on a besoin de $\lceil \log_2 5 \rceil = 3$ bits pour coder seulement 5 valeurs. Peut-on faire mieux?

Code de Golomb: codage du reste

- Codage du reste r

$$q = \left\lfloor \frac{n}{m} \right\rfloor, \quad r = n - qm$$

- Si m n'est pas une puissance de 2, on peut réduire le nombre de bits pour coder r.
- Pour représenter les $2^{\lceil \log_2 m \rceil} - m$ premières valeurs de r allant de $0, \cdots, 2^{\lceil \log_2 m \rceil} - m - 1$ on utilise une représentation binaire de longueur $\lfloor \log_2 m \rfloor$.
- Pour le reste des valeurs de r allant de $2^{\lceil \log_2 m \rceil} - m, \cdots, m - 1$ on code $r + 2^{\lceil \log_2 m \rceil} - m$ avec une représentation binaire de longueur $\lceil \log_2 m \rceil$.

Exemple du codage du reste pour m=5

- Encodage du premier ensemble de valeur de r:
  - Le premier ensemble de r est composé des entiers suivant:

$$r \in \{0, \cdots, 2^{\lceil \log_2 m \rceil} - m - 1\} = \{0, \cdots, 2^{\lceil \log_2 5 \rceil} - 5 - 1\}$$

$$r \in \{0, \cdots, 8 - 5 - 1\} = \{0, \cdots, 2\}$$

  - Représentation binaire de longueur $\lfloor \log_2 m \rfloor = \lfloor \log_2 5 \rfloor = 2$
  - $r=0=00_2$, $r=1=01_2$, $r=2=10_2$
- Encodage du deuxième ensemble de valeur de r:
  - Le reste des valeur de r forme l'ensemble suivant:

$$r \in \{2^{\lceil \log_2 m \rceil} - m, \cdots, m - 1\} = \{2^3 - 5, \cdots, 4\} = \{3, 4\}$$

  - Représentation binaire de longueur $\lceil \log_2 m \rceil = \lceil \log_2 5 \rceil = 3$
  - Codage de $r + 2^{\lceil \log_2 m \rceil} - m = r + 2^{\lceil \log_2 5 \rceil} - 5 = r + 3$
  - $r=3 \rightarrow r+3=6=110_2$, $r=4 \rightarrow r+3=7=111_2$

Exemple du codage de Golomb pour m=5

- Encodage du quotient avec un code unaire
  - q→q fois '1' et un '0'
- Encodage du reste:
  - $r=0=00_2$, $r=1=01_2$, $r=2=10_2$
  - $r=3 \rightarrow r+3=6=110_2$, $r=4 \rightarrow r+3=7=111_2$

$$q = \left\lfloor \frac{n}{m} \right\rfloor$$

$$r = n - qm$$

n	q	r	Mot-code
0	0	0	000
1	0	1	001
2	0	2	010
3	0	3	0110
4	0	4	0111
5	1	0	1000
6	1	1	1001
7	1	2	1010

n	q	r	Mot-code
8	1	3	10110
9	1	4	10111
10	2	0	11000
11	2	1	11001
12	2	2	11010
13	2	3	110110
14	2	4	110111
15	3	0	111000

Optimalité du codage de Golomb

- Cas d'optimalité
  - Le codage de Golomb est optimal pour le modèle de distribution géométrique:

$$P(n) = p^{n-1}(1-p)$$

  - Le choix du paramètre m assurant l'optimalité dans cette situation est déterminé par:

$$m = \left\lceil -\frac{1}{\log_2 p} \right\rceil$$

- Exemple d'utilisation
  - Encodage de séquence binaire avec un comptage des bits identiques.
  - Run length encoding (RLE)

Exemple de code de Golomb (1)

- Enoncé
  - Soit la séquence binaire suivante à encoder:

11111011111111100111111111101101111111010111111

  - En comptant le nombre des '1' successifs, extraire une séquence d'entier.
  - Estimer la probabilités P d'apparition de 1 dans la séquence binaire.
  - Déduire une estimation de la probabilité de chaque entier.
  - Encoder cette séquence d'entier avec un code de Golomb dont le paramètre m est à déterminer.
  - Déterminer l'entropie de la source.
  - Déterminer l'efficacité de ce code.

Exemple de code de Golomb (2)

- Solution
  - Comptage des '1' successifs

11111011111111100111111111101101111111010111111

-----5, ---------9,0, ----------10,--2, -------7,-1,---6

  - Séquence à encoder: 5,9,0,10,2,7,1,6
  - P=40/47=0.85106
  - Pr(n=3)=Pr(bit1=1)x Pr(bit2=1)x Pr(bit3=1)x Pr(bit4=0)
  - Pr(n=3)= $p^3x(1-p)$→ Pr(n)= $p^3x(1-p)$ [modèle géométrique].
  - Paramètre

$$m = \left\lceil -\frac{1}{\log_2 p} \right\rceil = \left\lceil -\frac{1}{\log_2 0.85106} \right\rceil = 5$$

Exemple de code de Golomb (3)

- Solution [suite]
  - Séquence à encoder: 5,9,0,10,2,7,1,6
  - Code: 1000 10111 000 11000 010 1010 001 1001 → 31
  - Longueur moyenne du code: 31/47= 0.65957 bits
  - Entropie de la source:$-p\log_2 p-(1-p)\log_2(1-p)$=0.60718 bits
  - Efficacité du code=0.6071 /0.65957 =92.055 %

n	q	r	Mot-code
0	0	0	000
1	0	1	001
2	0	2	010
3	0	3	0110
4	0	4	0111
5	1	0	1000
6	1	1	1001
7	1	2	1010

n	q	r	Mot-code
8	1	3	10110
9	1	4	10111
10	2	0	11000
11	2	1	11001
12	2	2	11010
13	2	3	110110
14	2	4	110111
15	3	0	111000

Les codes de Rice

- Principe
  - Le code de Rice est une extension adaptative du code de Golomb.
  - Le code de Rice traite des séquences d'entiers positifs qui peuvent être obtenu par un prétraitement de données.
  - Le code de Rice traite la séquence d'entrée en considérant une subdivision en block de longueur J.
  - Plusieurs options d'encodage sont testés pour chacun des blocks.
  - L'option d'encodage qui résulte en un nombre minimal de bits est sélectionnée.
  - L'option sélectionnée est représentée par un identificateur encapsulé dans le code de chacun des blocks.

Implémentation du code de Rice dans le CCSDS

- CCSDS
  - Recommandation pour la compression sans perte pour le ***Consultative Commitee on Space Data Standards***.
- Algorithme
  - Prétraitement des données: c'est une étape de modélisation des données pour la suppression de la corrélation et la génération d'entiers non négatifs.
  - Propriété de la séquence: les petites valeurs ont une probabilité plus élevée que les grandes valeurs.

    i.e. Pr(1)=0.4 et Pr(15)=0.001.
  - Encodeur binaire: associe une séquence binaire à chaque symboles selon l'option d'encodage sélectionné.

Étape de prétraitement

- Prétraitement
  - Soit une séquence d'entrée formée par $\{y_i\}$.
  - Pour chaque y_i on génère une prédiction $\hat{y}_i$.
  - Un façon simple de prédiction consiste à prendre $\hat{y}_i = y_{i-1}$.
  - Calcul de l'erreur de prédiction: $d_i = y_i - \hat{y}_i$.
  - Si le modèle est pertinent les valeurs de d_i seront faibles.
  - Soient y_{max} et y_{min} les valeurs minimale et maximale de $\{y_i\}$.
  - On définit: $T_i = \min\{y_{max} - \hat{y}_i, \hat{y}_i - y_{min}\}$.
  - Conversion de la séquence d_i en une séquence positive x_i :

$$x_i = \begin{cases} 2d_i & 0 \le d_i \le T_i \\ 2|d_i| - 1 & -T_i \le d_i \le 0 \\ T_i + |d_i| & \text{sinon} \end{cases}$$

Découpage de la séquence et encodage

- Partitionnement en block de taille J
  - Découpage de la séquence positive x_i en block de taille J=16 tel spécifié dans la recommandation CCSDS.

$$x_i = \begin{cases} 2d_i & 0 \le d_i \le T_i \\ 2|d_i| - 1 & -T_i \le d_i \le 0 \\ T_i + |d_i| & \text{sinon} \end{cases}$$

  - La valeur de x_i est petite si la valeur de d_i l'est aussi.
  - Est comme d_i prend des valeurs faibles la plupart du temps (si le modèle est pertinent) alors x_i prend avec une grande probabilité des valeurs petites.
- Encodage
  - L'encodage de chacun des blocks de taille J se fait selon une parmi 4 options.

Options d'encodage (1)

- Séquence fondamentale
  - Code unaire: n→n x '0' + 1 '1' ou n→n x '1' + 1 '0'
- Options du symbole fractionné
  - Ce sont des options régies par un paramètre m: pour un représentation en k bits, la $m^{ième}$ option du symbole fractionné considère les m bits les moins significatifs (LSB) comme tels et auxquels se rajoute le code unaire des k-m bits restants les plus significatifs (MSB).
- Exemple
  - Encodage en 8 bits de 23 utilisant la $3^{ème}$ option

23→00010111: 3 LSB= 111

5 MSB restants= 00010=2 → code unaire 001

23→ 111001

Options d'encodage (2)

- Options du symbole fractionné
  - Le paramètre m est sélectionné selon la séquence x_i.
  - Un m élevé est utilisé pour une entropie élevée de la séquence.
- Options de l'extention seconde
- Option du block zero

Les codes Tunstall (1)

- Principe
  - Un code à longueur variable encode les lettres de l'alphabet utilisant des mots code de longueur différente suivant la règle suivante:
  - les mot-codes avec le moins de bits pour les lettres qui apparaissent fréquemment et les mots codes avec le plus de bits pour les lettres les plus fréquents.
  - Les codes de Tunstall font l'exception à cette règle: les mot-codes sont de longueurs égales mais ils représentent un nombre différent de lettre.
  - L'avantage principal du code Tunstal est que l'erreur ne se propage pas contrairement aux autres codes à longueur variable (Huffman).

Les codes Tunstall (2)

- Exemple
  - Soit le code de Tunstall à 2 bits suivant:

Séquence	Mot-codes
AAA	00
AAB	01
AB	10
B	11

  - Encodage de AAABAABAABAAA
  - AAABAABAABAAA
  - 0011010100

Les codes Tunstall (3)

- Conditions
  - Le design d'un code Tunstal répond à deux conditions:
    1. Toute séquence de sources doit pouvoir être représentée par une séquence de symbole apparaissant dans le code.
    2. Il faut maximiser le nombre moyen de symboles de source représentés par chaque mot code.

Les codes Tunstall (4)

- Conditions 1
  - Pour comprendre la première condition, considérons le code suivant:

Séquence	Mot-codes
AAA	00
ABA	01
AB	10
B	11

  - Encoder la séquence: AAABAABAABAAA
  - AAABAABAABAAA
  - 0011
  - On n'a pas de code pour ni A ni AA ni AAB

Algorithme deTunstall

- Tunstall présente un algorithme simple qui répond à ces deux conditions.
- Code à n-bits pour une source iid d'un alphabet de taille N.
- Le nombre de mot code est 2^n.
- On prend l'entrée de plus grande probabilité on la concatène avec les autres lettres de l'alphabet.
- La taille du code passe de N à N+(N-1).
- Pr(nouvelle entrée)= ∏Pr(lettres)
- Itérer la même procédure.
- Chaque fois la taille du code augmente de N-1
- Après K itération la taille du code: N+K(N-1)
- Condition d'arrêt: $N+(K+1)(N-1)>2^n$

Exemple: Algorithme deTunstall (1)

- Énoncé
  - Construire un code de Tunstall à 3 bits pour une source sans mémoire de l'alphabet $A=\{A, B, C\}$. $P(A)=0.6$, $P(B)=0.3$ et $P(C)=0.1$.
- Itération 1

Lettre	Probabilité	Code
A	0.60	000
B	0.30	001
C	0.10	010
		011
		100
		101
		110
		111

Exemple: Algorithme deTunstall (2)

- Énoncé
  - Construire un code de Tunstall à 3 bits pour une source sans mémoire de l'alphabet $A=\{A, B, C\}$. $P(A)=0.6$, $P(B)=0.3$ et $P(C)=0.1$.
- Itération 2

Lettre	Probabilité	Code
B	0.30	000
C	0.10	001
AA	0.36	010
AB	018	011
AC	0.06	100
		101
		110
		111

Exemple: Algorithme deTunstall (3)

- Énoncé
  - Construire un code de Tunstall à 3 bits pour une source sans mémoire de l'alphabet $A=\{A, B, C\}$. $P(A)=0.6$, $P(B)=0.3$ et $P(C)=0.1$.
- Itération 3

Lettre	Probabilité	Code
B	0.30	000
C	0.10	001
AB	018	010
AC	0.06	011
AAA	0.216	100
AAB	0.108	101
AAC	0.036	110
		~~111~~