Codage numérique : du nombre au pixel - Cours 3

Codes correcteurs d'erreurs

L1 – Université de Lorraine
B. Girau et N. de Rugy Altherre

Transparents disponibles sur la plateforme de cours en ligne

Codes détecteur d'erreurs

Technique de codage permettant de détecter des erreurs de transmission d'un message (binaire) sur une voie de communication (numérique) peu fiable.

Codes correcteurs d'erreurs

Technique de codage permettant de détecter et de corriger des erreurs de transmission d'un message (binaire) sur une voie de communication (numérique) peu fiable.

Utilité

- La transmission d'informations entre des systèmes numériques distants utilise des protocoles complexes et passe par des canaux de transmission imparfaits.
- Une détection d'erreur à la réception d'un message permet de savoir s'il faut demander à l'émetteur de renvoyer le message.
- Une correction automatique des erreurs permet de ne même pas avoir à demander le renvoi du message si des erreurs sont détectées (dans certaines limites).
- Exemple : La probabilité d'erreur sur une ligne téléphonique est de 10^{-7} (cela peut même atteindre 10^{-4}). Avec un taux d'erreur de 10^{-6} et une connexion à 1 Mo/s, en moyenne 8 bits erronés sont transmis chaque seconde.

Domaines d'utilisation

- communications classiques (radio, câble coaxial, fibre optique, etc.)
- supports de stockage (disques durs, mémoire RAM, etc.) : cf Error-Correcting Code Memory ou ECC Memory
- autres applications où l'intégrité des données est importante

Mémoire et erreurs

- On a mesuré un taux d'erreurs (bit flip) de l'ordre de 1 bit par heure à 1 bit par seconde pour 1 To de mémoire (suivant technologie).
- Ce taux a tendance à diminuer avec les nouvelles technologies ... mais comme la taille des mémoires augmente.
- Cause principale : rayonnement (surtout neutrons dus aux rayonnement cosmique), mais aussi fuite de charge avec les technologies de très grande densité.
- Rayonnement augmente avec l'altitude (x3.5 à 1500m, x3000 à 10000m), d'où la nécessité de prévenir les effets de telles erreurs, par exemple dans les avions.

Contrôle par redondance

Idée fondamentale des codes détecteurs et des codes correcteurs d'erreurs : augmenter le message transmis au moyen d'une fonction liée aux données à transmettre. Le code ainsi transmis contient des redondances d'information. Toute incohérence dans cette redondance permet de détecter un problème.

Exemple basique

On peut doubler chaque bit transmis : pour transmettre 01010101, on transmet en fait 0011001100110011.
En cas d'erreur simple, on détecte immédiatement un problème : 0011001100**01**0011.
S'il y a deux erreurs, ça dépend de leur emplacement ...

Principe général

- Chaque suite de bits à transmettre est augmentée par une autre suite de bits dite « de redondance » ou « de contrôle ».
- Pour chaque suite de k bits transmise, on ajoute r bits. On dit alors que l'on utilise un code $C(n, k)$ avec $n = k + r$.
- À la réception, les bits ajoutés permettent d'effectuer des contrôles.

Sommes de contrôle

Une solution simple (et optimale si on veut juste détecter les erreurs à moindre coût) est de faire suivre le message d'une somme de contrôle (checksum) calculée sur les données à transmettre.

Attention : le nombre d'erreurs ne doit pas dépasser un seuil (en général petit).

Cas simple : le bit de parité

On rajoute à la fin des données binaires à transmettre un bit égal à 1 si la somme des bits à transmettre est impaire, 0 sinon.

Attention : ne pas confondre avec la parité du nombre représenté : 01000000 est pair, mais la somme des bits est impaire.

Exemples d'utilisation

ASCII : les caractères codés sur 7 bits étaient fréquemment accompagnés d'un bit de parité (et donc transmis octet par octet).

Cas plus complexe

Le bit de parité est un exemple de code dit linéaire. D'autres fonctions linéaires peuvent être envisagées pour transformer le message d'origine en sa version redondante.

Il existe différentes versions plus évoluées de la somme de contrôle : CRC (cyclic redundancy code), MD5 (message digest 5), BCH, etc., qui ajoutent des propriétés fortes (sécurité, correction, etc.) au prix d'un surcoût plus élevé qu'une simple somme de contrôle.

Un code de Hamming est un code utilisant des sommes de contrôle multiples.

Code de Hamming

Permet la détection et la correction automatique d'une erreur si elle ne porte que sur une lettre du message.

Un code de Hamming est parfait : pour une longueur de code donnée, il n'existe pas d'autre code plus compact ayant la même capacité de correction.

Structure d'un code de Hamming

- les m bits du message à transmettre et les n bits de contrôle de parité.
- longueur totale : $2^n - 1$
- longueur du message : $m = (2^n - 1) - n$
- on parle de code x-y : x est la longueur totale du code $(n + m)$ et y la longueur du message (m)

Exemples

- un code 7-4 a un coefficient d'efficacité de $4/7 = 57\ \%$
- un code 15-11 a un coefficient d'efficacité de $11/15 = 73\ \%$
- un code 31-26 a un coefficient d'efficacité de $26/31 = 83\ \%$

Structure d'un code de Hamming (suite)

- les bits de contrôle C_i sont en position 2^i pour $i = 0, 1, 2, \ldots$
- les bits du message D_j occupent le reste du message

Exemple : structure d'un code de Hamming 7-4

position	7	6	5	4	3	2	1
bit D/C	D_3	D_2	D_1	C_2	D_0	C_1	C_0

Structure d'un code de Hamming (fin)

Le bit de contrôle C_i est le bit de parité de tous les D_j tels que le bit i est égal à 1 dans l'écriture binaire de la position de D_j.

Remarques :

- dans un code de Hamming $(n+m)-m$, chaque C_i contrôle ainsi le même nombre $2^{n-1}-1$ de bits de données à transmettre
- chaque D_j est ainsi contrôlé par autant de C_i qu'il y a de 1 dans l'écriture binaire de la position de D_j dans le code
- deux D_j distincts ne sont jamais contrôlés par exactement la même liste de C_i

Exemple : bits de contrôle dans un code de Hamming 7-4

position	position (binaire)	parité C_2	parité C_1	parité C_0
1 (C_0)	001			X
2 (C_1)	010		X	
3 (D_0)	011		X	X
4 (C_2)	100	X		
5 (D_1)	101	X		X
6 (D_2)	110	X	X	
7 (D_3)	111	X	X	X

Donc C_0 est le bit de parité de D_0, D_1 et D_3, C_1 est le bit de parité de D_0, D_2 et D_3, et C_2 est le bit de parité de D_1, D_2 et D_3

Exemple : bits de contrôle dans un code de Hamming 7-4

On souhaite envoyer le message 1011.

position	7	6	5	4	3	2	1
bit D/C	1	0	1	C_2	1	C_1	C_0

- C_0 vaut 1 : $1+1+1$ (les bits d'indices 7, 5, 3).
- C_1 vaut 0 : $1+0+1$ (les bits d'indices 7, 6, 3).
- C_2 vaut 0 : $1+0+1$ (les bits d'indices 7, 6, 5).

position	7	6	5	4	3	2	1
bit D/C	1	0	1	0	1	0	1

On envoie donc 1010101.

Retrouver l'erreur dans un mot de Hamming

Si les bits de contrôle de parité C_2, C_1, C_0 ont tous la bonne valeur, il n'y a pas d'erreurs ; sinon les bits de contrôle erronés indique la position de l'erreur entre 1 et 7 : l'unique bit corrompu est celui dont la position est codée par $f_2 f_1 f_0$, où f_i est égal à 1 si et seulement si C_i ne correspond pas à la parité calculée.

Exemple : erreur dans un code de Hamming 7-4

On souhaite envoyer le message 1011

...

on envoie 1010101, mais

...

on reçoit 1110101

Exemple : erreur dans un code de Hamming 7-4

On a reçu 1110101

position	7	6	5	4	3	2	1
bit D/C	1	1	1	0 vs C_2	1	0 vs C_1	1 vs C_0

- C_0 devrait être égal à $1 : 1+1+1$ (les bits d'indices 7, 5, 3), donc $f_0 = 0$.
- C_1 devrait être égal à $1 : 1+1+1$ (les bits d'indices 7, 6, 3), donc $f_1 = 1$.
- C_2 devrait être égal à $1 : 1+1+1$ (les bits d'indices 7, 6, 5), donc $f_2 = 1$.

Le bit faux est en position $110 = 6$.

Ca marche même si le bit faux est lui-même un bit de contrôle.

Exemple : erreur de contrôle dans un code de Hamming 7-4

On reçoit 1010111

position	7	6	5	4	3	2	1
bit D/C	1	0	1	0 vs C_2	1	1 vs C_1	1 vs C_0

- C_0 devrait être égal à $1 : 1 + 1 + 1$ (les bits d'indices 7, 5, 3), donc $f_0 = 0$.
- C_1 devrait être égal à $0 : 1 + 0 + 1$ (les bits d'indices 7, 6, 3), donc $f_1 = 1$.
- C_2 devrait être égal à $0 : 1 + 0 + 1$ (les bits d'indices 7, 6, 5), donc $f_2 = 0$.

Le bit faux est en position $010 = 2$, c'est le second bit de contrôle.