Codes correcteurs d'erreurs

Bruno Grenet

2017 — 2018

Introduction

But Transmettre de l'information au travers d'un canal bruité, ce qui nécessite de savoir repérer et corriger les erreurs

Exemple Alice envoie le message `Bonjour` à Bob. Bob reçoit `Bonjoir`. Comment faire pour que Bob sache que le message originel est `Bonjour` et non `Bonsoir` ?

Idée Ajouter de la redondance. Par exemple, si Alice envoie `BBBooonnnjjjoooouuurrr` et que Bob reçoit `BBBooonnnjjjoooouuirrr`, il peut par vote de majorité corriger la réception pour en déduire le message.

Objectifs

1. Ajouter le moins de redondance possible (pour limiter le poids des messages à transmettre) tout en sachant corriger un maximum d'erreurs.
2. S'assurer que les complexités des opérations d'encodage et décodage soit les plus faibles possibles.

1 Le code de Hamming

Contexte

— Messages : mots de 4 bits
— But : pouvoir corriger 1 erreur (à coup sûr)

On peut pour cela tripler le message comme dans l'exemple introductif[ˆrmq]. Ainsi, on encodera les message $x_1x_2x_3x_4$ par $x_1x_2x_3x_4x_1x_2x_3x_4x_1x_2x_3x_4$. Le but est de réduire la redondance nécessaire. On peut déjà remarquer que la troisième copie du message n'est pas nécessaire : on peut la remplacer par 1 bit de parité $p = x_1 \oplus x_2 \oplus x_3 \oplus x_4$. On est alors encore capable de corriger une erreur, où qu'elle soit (exercice).

Définition Le *code de Hamming* consiste à encoder le message $x_1x_2x_3x_4$ sous la forme

$$x_1x_2x_3x_4p_1p_2p_3$$

où les *bits de parité* p_1, p_2 et p_3 sont définis par

$$\begin{aligned} p_1 &= x_1 \oplus x_2 \oplus x_3 \\ p_2 &= x_1 \oplus x_2 \oplus x_4 \\ p_3 &= x_1 \oplus x_3 \oplus x_4 \end{aligned}$$

La définition du code peut être interprêtée sur la figure 1 : un mot est valide (*mot de code*) si la somme *modulo* 2 des quatre bits de chaque cercle vaut 0.

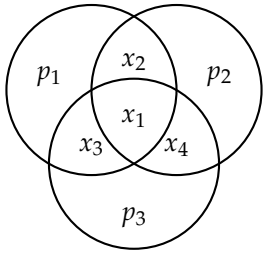

FIGURE 1 – Illustration du code de Hamming

Pour décoder, il suffit de calculer les trois sommes correspondant à chaque cercle :
— Si toutes les sommes valent 0, le mot est correct ;
— Sinon, le bit faux est celui qui se trouve à l'intersection des cercles dont la somme vaut 1 (et qui n'est pas dans le ou les cercles dont la somme vaut 0).

Par exemple, si la somme du cercle du bas vaut 1 et les deux autres 0, c'est le bit p_3 qui est faux. Si les trois sommes valent 1, c'est le bit x_1 qui est faux. On peut montrer la proposition suivante (exercice).

Proposition Si un mot du code de Hamming est reçu avec au plus 1 erreur, il existe une unique manière de le décoder (c'est-à-dire qu'il existe un unique mot de code à distance 1 du mot reçu).

Définition La *distance de Hamming* entre deux mots u et v (de mêmes tailles) sur un alphabet Σ est le nombre d'indices i tels que $u_i \neq v_i$. On la note $d(u, v)$.

Proposition La distance de Hamming entre deux mots de code est ≥ 3.

Preuve. On considère deux mots de code distincts $\hat{x} = x_1x_2x_3x_4p_1p_2p_3$ et $\hat{y} = y_1y_2y_3y_4q_1q_2q_3$. On note x et y les quatre premiers bits respectifs de ces deux mots. Le résultat se prouve par étude de cas sur la distance entre x et y.
— $d(x, y) \geq 3$: Alors $d(\hat{x}, \hat{y}) \geq 3$ également.
— $d(x, y) = 2$: Il y a deux cas à considérer.

- Soit $x_1 = y_1$, auquel cas il existe deux indices $2 \leq i < j \leq 4$ tels que $x_i \neq y_i$ et $x_j \neq y_j$. On note k le dernier indice. Alors $x_1 \oplus x_i \oplus x_k \neq y_1 \oplus y_i \oplus y_k$ puisque $x_1 = y_1$ et $x_k = y_k$. Les bits de parité correspondant sont donc différents et $d(\hat{x}, \hat{y}) \geq 3$ (et même ≥ 4 en considérant la même somme avec l'indice j).
- Sinon, il existe un indice i tel que $x_i \neq y_i$; les deux autres indices sont notés j et k. Alors $x_1 \oplus x_j \oplus x_k \neq y_1 \oplus y_j \oplus y_k$ et $d(\hat{x}, \hat{y}) \geq 3$.

— $d(x, y) = 1$: On considère encore deux cas.
- Si $x_1 = y_1$, on note i l'indice tel que $x_i \neq y_i$ et j et k les deux autres. Alors les deux bits de parité dans lesquels intervient l'indice i sont différents entre $\hat{x}$ et $\hat{y}$. D'où $d(\hat{x}, \hat{y}) \geq 3$.
- Finalement, si $x_1 \neq y_1$, tous les bits de parité diffèrent entre $\hat{x}$ et $\hat{y}$ et $d(\hat{x}, \hat{y}) \geq 3$.

Remarque Cette proposition permet de démontrer qu'on peut toujours corriger une erreur : puisque deux mots de code sont à distance au moins 3, quand on s'est éloigné à distance 1 d'un mot de code, on ne peut pas être à distance 1 d'un autre mot de code. Autrement dit, le décodage est bien unique.

2 Formalisation

Définition Un $(n, M, d)_\Sigma$-code est un sous-ensemble $C \subset \Sigma^n$ de taille M tel que $d = \min\limits_{\substack{x,y \in C \\ x \neq y}} d(x, y)$.

Vocabulaire
- n : longueur du code ;
- M : taille du code ;
- d : distance (minimale) du code ;
- $k = \log_{|\Sigma|} M$: dimension du code ;
- $R = k/n$: ratio du code ;
- Les éléments de C sont appelés les *mots de code*.
- On remplace parfois l'indice Σ par la taille de l'alphabet Σ. En particulier, si $\Sigma = \{0, 1\}$, on utilise en général la notation $(n, M, d)_2$.
- *Décoder* un mot $u \in \Sigma^n$ signifie calculer un mot de code $c \in C$ le plus proche de u.

Exemple Le code de Hamming est un $(7, 16, 3)_2$ code, de dimension $k = \log_2(16) = 4$ et de ratio $4/7$.

Remarque La définition de code correcteur ne parle pas de l'espace *des messages* (les mots de longueur 4 dans le cas du code de Hamming) ou de l'encodage. On ne considère que les mots encodés. La seule information reliée à cet espace des messages est la dimension, qui correspond (plus ou moins) à la longueur des messages qu'on encode. Une autre façon de le voir est que la dimension est le nombre de *bits d'information* dans les mots de code.

Proposition Soit C un code. Alors les assertions suivantes sont équivalentes.

i. La distance de C est d.

ii. Si d est impair, C permet de corriger $\frac{d-1}{2}$ erreurs.
iii. C permet de détecter $d-1$ erreurs.
iv. C permet de corriger $d-1$ *effacements* (c'est-à-dire que $d-1$ symboles sont remplacés par un symbole spécial ?).

Preuve. Dans la preuve, c et c' désignent des mots de code distincts, et u un mot quelconque. On note que si C a distance d, alors $d(c,c') \geq d$ par définition. Réciproquement, si C a une distance $< d$, il existe deux mots de code c et c' tels que $d(c,c') \leq d-1$.

— $i \iff ii$: Si C a distance d, on ne peut pas avoir $d(u,c) \leq (d-1)/2$ et $d(u,c') \leq (d-1)/2$ car l'inégalité triangulaire impliquerait $d \leq d(c,c') \leq d(u,c) + d(u,c') \leq d-1$. Réciproquement, si $d(c,c') \leq d-1$, alors il existe un mot u *entre* c *et* c', à égale distance $(d-1)/2$ de chacun : on ne peut pas le décoder de manière unique.

— $i \iff iii$: Si C a distance d et $d(u,c) = d-1$, alors u ne peut pas être lui même un mot de code : on détecte la présence d'erreurs. Réciproquement si $d(c,c') \leq d-1$, on peut faire $d-1$ erreurs et tomber sur c' : on ne détecte pas la présence d'erreurs.

— $i \iff iv$: Si C a distance d et que u a $d-1$ symboles effacés, et que les symboles non effacés de u étaient égaux à la fois à ceux de c et de c', alors c et c' seraient à distance $d-1$: on peut donc décoder u de manière unique. Réciproquement, si $d(c,c') = d-1$ et qu'on efface les symboles qui diffèrent, on ne peut pas décoder.

3 Bornes combinatoires

Théorème (borne de Hamming) Soit C un $(n, M, d)_\Sigma$-code avec $|\Sigma| = q$. Alors

$$M \cdot V_q(n, \lfloor \frac{d-1}{2} \rfloor) \leq q^n$$

où pour tout $t \geq 0$, $V_q(n,t)$ désigne le *volume de la boule de rayon* t *dans* Σ^n, c'est-à-dire le nombre de mots à distance au plus t d'un mot fixé, et vérifie

$$V_q(n,t) = \sum_{i=0}^{t} \binom{n}{i} (q-1)^i.$$

Preuve. On montre d'une part que $MV \leq q^n$ où V est le volume de la boule de rayon $\lfloor (d-1)/2 \rfloor$. Pour cela, on considère les M mots de codes. Puisque qu'on peut corriger jusqu'à $\lfloor (d-1)/2 \rfloor$ erreurs [1], les boules de ce rayon centrées en chacun des mots de code ont une intersection vide. Ainsi, on peut mettre M boules disjointes de volume V dans Σ^n. D'où le résultat (puisque $|\Sigma^n| = q^n$).

Pour calculer $V_q(n,t)$, on considère le nombre de mots de Σ^n à distance exactement i (≥ 0) d'un mot u donné : un tel mot est obtenu en choisissant i symboles à modifier parmi les n symboles de u, puis pour chacun en choisissant l'un des symboles de l'alphabet, différent du symbole dans u. Il y a donc $\binom{n}{i}(q-1)^i$ mots à distance exactement i de u. Finalement, le volume de la boule centrée en u et de rayon t est le nombre de mots à distance *au plus* t de u, d'où le résultat.

1. Remarque : on n'a montré ce résultat que pour d impair. On peut facilement le montrer pour d pair (exercice).

Théorème (borne de Singleton) Soit C un $(n, M, d)_\Sigma$ code avec $|\Sigma| = q$. Alors $d \leq n - \log_q M + 1$.

Preuve. Soit $\ell = \lceil \log_q M \rceil - 1$. Alors $\ell > \log_q M$ d'où $q^\ell < M$. Comme il y a M mots de codes distincts, au moins deux d'entre eux coïncident sur leurs ℓ premières lettres (il n'y a que q^ℓ préfixes de longueur ℓ distincts). Ces deux mots de codes sont à distance $\leq n - \ell$, d'où le résultat.

Définitions

— Un code est dit *parfait* s'il atteint la borne de Hamming.
— Un code est dit *MDS* (pour *Maximum Distance Separable*) s'il atteint la borne de Singleton.

Remarques

— Les deux bornes sont incomparables : il existe des codes parfaits mais non MDS (ex : le code de Hamming [2]) et des codes MDS mais non parfaits (ex : le code de Reed-Solomon [3]).
— De nombreuses autres bornes, souvent incomparables entre elles, existent.

2. Exercice : démontrer que le code de Hamming est parfait mais pas MDS.
3. On verra ce code dans la suite.