Codes correcteurs d'erreurs

Jérémy Le Borgne

3 mars 2016





Signaux numériques

On transmet beaucoup d'informations sous forme *numérique*, c'est-à-dire sous forme de suites de 0 et de 1.

Signaux numériques

On transmet beaucoup d'informations sous forme *numérique*, c'est-à-dire sous forme de suites de 0 et de 1.

11000101011101

Signaux numériques

On transmet beaucoup d'informations sous forme *numérique*, c'est-à-dire sous forme de suites de 0 et de 1.

11000101011101

C'est sous cette forme que les informations transitent dans un ordinateur : ce sont des courants électriques qui circulent (=1) ou pas (=0).

Transmission du signal

Un signal numérique se propage généralement dans :

Transmission du signal

Un signal numérique se propage généralement dans :

- un circuit imprimé

Transmission du signal

Un signal numérique se propage généralement dans :

- un circuit imprimé
- un câble

Transmission du signal

Un signal numérique se propage généralement dans :

- un circuit imprimé
- un câble
- le vide

Transmission du signal

Un signal numérique se propage généralement dans :

- un circuit imprimé
- un câble
- le vide

Il peut y avoir des altérations de l'information :

Transmission du signal

Un signal numérique se propage généralement dans :

- un circuit imprimé
- un câble
- le vide

Il peut y avoir des altérations de l'information :

- $0 \leftrightarrow 1$

Transmission du signal

Un signal numérique se propage généralement dans :

- un circuit imprimé
- un câble
- le vide

Il peut y avoir des altérations de l'information :

- $0 \leftrightarrow 1$
- paquets perdus

Détecter et corriger les erreurs

Pouvez-vous proposer une méthode pour détecter (et si possible corriger) une erreur de transmission dans un signal numérique ?

Détecter et corriger les erreurs

Pouvez-vous proposer une méthode pour détecter (et si possible corriger) une erreur de transmission dans un signal numérique ?
Message à transmettre :

1001

Il faudra probablement transmettre davantage d'informations...

Un exemple plus élaboré

- Choisissez un nombre entre 0 et 15.
- Répondez à chacune des questions suivantes à propos de votre nombre (vous pouvez mentir une fois) :

Un exemple plus élaboré

- Choisissez un nombre entre 0 et 15.
- Répondez à chacune des questions suivantes à propos de votre nombre (vous pouvez mentir une fois) :

1. Est-il ≥ 8 ?

Un exemple plus élaboré

- Choisissez un nombre entre 0 et 15.
- Répondez à chacune des questions suivantes à propos de votre nombre (vous pouvez mentir une fois) :

2. Est-il dans $\{4, 5, 6, 7, 12, 13, 14, 15\}$?

Un exemple plus élaboré

- Choisissez un nombre entre 0 et 15.
- Répondez à chacune des questions suivantes à propos de votre nombre (vous pouvez mentir une fois) :

3. Est-il dans $\{2, 3, 6, 7, 10, 11, 14, 5\}$?

Un exemple plus élaboré

- Choisissez un nombre entre 0 et 15.
- Répondez à chacune des questions suivantes à propos de votre nombre (vous pouvez mentir une fois) :

4. Est-il impair ?

Un exemple plus élaboré

- Choisissez un nombre entre 0 et 15.
- Répondez à chacune des questions suivantes à propos de votre nombre (vous pouvez mentir une fois) :

5. Est-il dans $\{1, 2, 4, 7, 9, 10, 12, 15\}$?

Un exemple plus élaboré

- Choisissez un nombre entre 0 et 15.
- Répondez à chacune des questions suivantes à propos de votre nombre (vous pouvez mentir une fois) :

6. Est-il dans $\{1, 2, 5, 6, 8, 11, 12, 15\}$?

Un exemple plus élaboré

- Choisissez un nombre entre 0 et 15.
- Répondez à chacune des questions suivantes à propos de votre nombre (vous pouvez mentir une fois) :

7. Est-il dans $\{1, 3, 4, 6, 8, 10, 13, 15\}$?

Comment ça marche ?

Les 4 premières questions déterminent la réponse si on n'a pas menti.

Comment ça marche ?

Les 4 premières questions déterminent la réponse si on n'a pas menti.
Ecrivons le nombre n en base 2 :

Comment ça marche ?

Les 4 premières questions déterminent la réponse si on n'a pas menti.
Ecrivons le nombre n en base 2 :

$$a_3 a_2 a_1 a_0$$

Comment ça marche ?

Les 4 premières questions déterminent la réponse si on n'a pas menti.
Ecrivons le nombre n en base 2 :

$$a_3 a_2 a_1 a_0$$

$$a_3 = 1$$

Comment ça marche ?

Les 4 premières questions déterminent la réponse si on n'a pas menti.
Ecrivons le nombre n en base 2 :

$$a_3 a_2 a_1 a_0$$

$$a_3 = 1 \quad \Leftrightarrow \qquad\qquad n \geq 8$$

Comment ça marche ?

Les 4 premières questions déterminent la réponse si on n'a pas menti.
Ecrivons le nombre n en base 2 :

$$a_3 a_2 a_1 a_0$$

$$\begin{array}{l} a_3 = 1 \\ a_2 = 1 \end{array} \quad \Leftrightarrow \quad n \geq 8$$

Comment ça marche ?

Les 4 premières questions déterminent la réponse si on n'a pas menti.
Ecrivons le nombre n en base 2 :

$$a_3 a_2 a_1 a_0$$

$$\begin{aligned} a_3 = 1 &\Leftrightarrow n \geq 8 \\ a_2 = 1 &\Leftrightarrow n \equiv 4, 5, 6, 7 \pmod 8 \end{aligned}$$

Comment ça marche ?

Les 4 premières questions déterminent la réponse si on n'a pas menti.
Ecrivons le nombre n en base 2 :

$$a_3 a_2 a_1 a_0$$

$$\begin{array}{lcl} a_3 = 1 & \Leftrightarrow & n \geq 8 \\ a_2 = 1 & \Leftrightarrow & n \equiv 4, 5, 6, 7 \pmod 8 \\ a_1 = 1 & & \end{array}$$

Comment ça marche ?

Les 4 premières questions déterminent la réponse si on n'a pas menti.
Ecrivons le nombre n en base 2 :

$$a_3 a_2 a_1 a_0$$

$$\begin{aligned} a_3 = 1 &\Leftrightarrow & n &\geq 8 \\ a_2 = 1 &\Leftrightarrow & n &\equiv 4, 5, 6, 7 \pmod 8 \\ a_1 = 1 &\Leftrightarrow & n &\equiv 2, 3 \pmod 4 \end{aligned}$$

Comment ça marche ?

Les 4 premières questions déterminent la réponse si on n'a pas menti.
Ecrivons le nombre n en base 2 :

$$a_3 a_2 a_1 a_0$$

$$\begin{array}{lcc} a_3 = 1 & \Leftrightarrow & n \geq 8 \\ a_2 = 1 & \Leftrightarrow & n \equiv 4, 5, 6, 7 \pmod 8 \\ a_1 = 1 & \Leftrightarrow & n \equiv 2, 3 \pmod 4 \\ a_0 = 1 & & \end{array}$$

Comment ça marche ?

Les 4 premières questions déterminent la réponse si on n'a pas menti.
Ecrivons le nombre n en base 2 :

$$a_3 a_2 a_1 a_0$$

$$\begin{aligned}
a_3 = 1 &\Leftrightarrow n \geq 8 \\
a_2 = 1 &\Leftrightarrow n \equiv 4, 5, 6, 7 \pmod 8 \\
a_1 = 1 &\Leftrightarrow n \equiv 2, 3 \pmod 4 \\
a_0 = 1 &\Leftrightarrow n \equiv 1 \pmod 2
\end{aligned}$$

Comment ça marche ?

Les 4 premières questions déterminent la réponse si on n'a pas menti.
Ecrivons le nombre n en base 2 :

$$a_3 a_2 a_1 a_0$$

$$\begin{aligned}
a_3 = 1 &\Leftrightarrow n \geq 8 \\
a_2 = 1 &\Leftrightarrow n \equiv 4, 5, 6, 7 \pmod 8 \\
a_1 = 1 &\Leftrightarrow n \equiv 2, 3 \pmod 4 \\
a_0 = 1 &\Leftrightarrow n \equiv 1 \pmod 2
\end{aligned}$$

Ainsi, a_3, a_2, a_1, a_0 sont les réponses aux 4 premières questions (avec 1 pour oui et 0 pour non).

Comment ça marche ?

Les informations supplémentaires servent à détecter la position d'une éventuelle erreur.

Comment ça marche ?

Les informations supplémentaires servent à détecter la position d'une éventuelle erreur.
Idée : pour chaque choix d'un nombre de départ, il y a 8 façons différentes de répondre aux questions.

Comment ça marche ?

Les informations supplémentaires servent à détecter la position d'une éventuelle erreur.
Idée : pour chaque choix d'un nombre de départ, il y a 8 façons différentes de répondre aux questions.
Pour chacune de ces 8 réponses possibles, on veut retrouver celle qui correspond à une réponse sans mensonges.

Comment ça marche ?

Les informations supplémentaires servent à détecter la position d'une éventuelle erreur.
Idée : pour chaque choix d'un nombre de départ, il y a 8 façons différentes de répondre aux questions.
Pour chacune de ces 8 réponses possibles, on veut retrouver celle qui correspond à une réponse sans mensonges.
On cherche la réponse sans mensonges « la plus proche »de celle qu'on obtient.

Distances

Soit X un ensemble. On appelle *distance* sur X une application :

$$d \ : \ X \times X \to \mathbf{R}$$

telle que :

Distances

Soit X un ensemble. On appelle *distance* sur X une application :

$$d \ : \ X \times X \to \mathbf{R}$$

telle que :

- $\forall x, y \in X,\ d(x,y) = 0 \Leftrightarrow x = y$;
- $\forall x, y \in X,\ d(x,y) = d(y,x)$;
- $\forall x, y, z \in X,\ d(x,z) \leq d(x,y) + d(y,z)$.

Distances

Soit X un ensemble. On appelle *distance* sur X une application :

$$d \ : \ X \times X \to \mathbf{R}$$

telle que :

- $\forall x, y \in X,\ d(x,y) = 0 \Leftrightarrow x = y$;
- $\forall x, y \in X,\ d(x,y) = d(y,x)$;
- $\forall x, y, z \in X,\ d(x,z) \leq d(x,y) + d(y,z)$.

Exemples :

- $X = \mathbf{R},\ d(x,y) = |x - y|$
- $X = \mathcal{C}^0([0,1], \mathbf{R}),\ d(f,g) = \sup_{x \in [0,1]} |f(x) - g(x)|$

Distance de Hamming

On note $X = \{0,1\}^n$ l'ensemble des séquences de 0 et de 1 de longueur n fixée :

$$x = (x_0, x_1, \ldots, x_{n-1}).$$

On définit $d : X \times X \to \mathbf{R}$ en disant que $d(x,y)$ est le nombre d'indices i tels que $x_i \neq y_i$.

Théorème

L'application d ainsi définie est une une distance sur X, appelée distance de Hamming.

Codes correcteurs

Soit $C \subset X$ un sous-ensemble de $X = \{0, 1\}^n$.

Codes correcteurs

Soit $C \subset X$ un sous-ensemble de $X = \{0,1\}^n$.
L'ensemble X représente l'ensemble de tous les « mots »possibles, et l'ensemble C est l'ensemble des mots que l'on s'autorise à envoyer.

Codes correcteurs

Soit $C \subset X$ un sous-ensemble de $X = \{0,1\}^n$.
L'ensemble X représente l'ensemble de tous les « mots »possibles, et l'ensemble C est l'ensemble des mots que l'on s'autorise à envoyer.
Les altérations du signal font que le message reçu n'est pas forcément un élément de C.

Codes correcteurs

Soit $C \subset X$ un sous-ensemble de $X = \{0,1\}^n$.
L'ensemble X représente l'ensemble de tous les « mots »possibles, et l'ensemble C est l'ensemble des mots que l'on s'autorise à envoyer.
Les altérations du signal font que le message reçu n'est pas forcément un élément de C.

Distance minimale et correction d'erreurs

Soit

$$\delta = \min\{d(x, y) \mid x, y \in C,\ x \neq y\}$$

On appelle δ la distance minimale du code.

Distance minimale et correction d'erreurs

Soit

$$\delta = \min\{d(x, y) \mid x, y \in C,\ x \neq y\}$$

On appelle δ la distance minimale du code.
On suppose que le canal de transmission fait *au plus* t erreurs.

Distance minimale et correction d'erreurs

Soit

$$\delta = \min\{d(x, y) \mid x, y \in C,\ x \neq y\}$$

On appelle δ la distance minimale du code.
On suppose que le canal de transmission fait *au plus* t erreurs.

Théorème

Si $t < 2\delta$, alors pour tout message transmis par le canal, il existe un unique mot de C qui est à distance du Hamming $\leq t$ du message transmis.

Distance minimale et correction d'erreurs

Soit

$$\delta = \min\{d(x, y) \mid x, y \in C,\ x \neq y\}$$

On appelle δ la distance minimale du code.
On suppose que le canal de transmission fait *au plus* t erreurs.

Théorème

Si $t < 2\delta$, alors pour tout message transmis par le canal, il existe un unique mot de C qui est à distance du Hamming $\leq t$ du message transmis.

Cet unique mot était donc le mot de départ !

Une vision plus algébrique

On considère le corps $\mathbf{K} = \mathbf{Z}/2\mathbf{Z}$.

Une vision plus algébrique

On considère le corps $\mathbf{K} = \mathbf{Z}/2\mathbf{Z}$. En tant qu'ensemble, c'est juste $\{0, 1\}$, mais il est muni d'une addition et d'une multiplication.

Une vision plus algébrique

On considère le corps $\mathbf{K} = \mathbf{Z}/2\mathbf{Z}$. En tant qu'ensemble, c'est juste $\{0, 1\}$, mais il est muni d'une addition et d'une multiplication.
$X = \mathbf{K}^n$ est un $\mathbf{K}$-espace vectoriel de dimension n.

Une vision plus algébrique

On considère le corps $\mathbf{K} = \mathbf{Z}/2\mathbf{Z}$. En tant qu'ensemble, c'est juste $\{0, 1\}$, mais il est muni d'une addition et d'une multiplication.
$X = \mathbf{K}^n$ est un $\mathbf{K}$-espace vectoriel de dimension n.
On peut chercher les codes C qui sont des sous-espaces vectoriels de X.

Une vision plus algébrique

On considère le corps $\mathbf{K} = \mathbf{Z}/2\mathbf{Z}$. En tant qu'ensemble, c'est juste $\{0, 1\}$, mais il est muni d'une addition et d'une multiplication.
$X = \mathbf{K}^n$ est un $\mathbf{K}$-espace vectoriel de dimension n.
On peut chercher les codes C qui sont des sous-espaces vectoriels de X.

Théorème

Soit $X = \mathbf{K}^n$ et $C \subset K$ est un sous-espace vectoriel de dimension k. Soit δ la distance minimale de C. Alors

$$\delta + k \leq n + 1.$$

Exemple : Codes de Reed-Solomon

Soit $n \in \mathbf{N}$, $k \leq n$, on se donne $(x_1, \ldots, x_n) \in \mathbf{K}^n$ deux à deux distincts, et :

$$c : \begin{array}{ccc} \mathbf{K}[X]_{<k} & \to & \mathbf{K}^n \\ P & \mapsto & (P(x_1), \ldots, P(x_n)) \end{array}$$

Exemple : Codes de Reed-Solomon

Soit $n \in \mathbf{N}$, $k \leq n$, on se donne $(x_1, \ldots, x_n) \in \mathbf{K}^n$ deux à deux distincts, et :

$$c \; : \; \begin{array}{ccc} \mathbf{K}[X]_{<k} & \to & \mathbf{K}^n \\ P & \mapsto & (P(x_1), \ldots, P(x_n)) \end{array}$$

- $X = \mathbf{K}^n$, $C = \mathrm{Im}(c)$ (avec la distance de Hamming)

Exemple : Codes de Reed-Solomon

Soit $n \in \mathbf{N}$, $k \leq n$, on se donne $(x_1, \ldots, x_n) \in \mathbf{K}^n$ deux à deux distincts, et :

$$c : \begin{array}{ccc} \mathbf{K}[X]_{<k} & \to & \mathbf{K}^n \\ P & \mapsto & (P(x_1), \ldots, P(x_n)) \end{array}$$

- $X = \mathbf{K}^n$, $C = \mathrm{Im}(c)$ (avec la distance de Hamming)
- Si $c(P)$ a au moins k coordonnées nulles, alors $P = 0$

Exemple : Codes de Reed-Solomon

Soit $n \in \mathbf{N}$, $k \leq n$, on se donne $(x_1, \ldots, x_n) \in \mathbf{K}^n$ deux à deux distincts, et :

$$c : \begin{array}{ccc} \mathbf{K}[X]_{<k} & \to & \mathbf{K}^n \\ P & \mapsto & (P(x_1), \ldots, P(x_n)) \end{array}$$

- $X = \mathbf{K}^n$, $C = \mathrm{Im}(c)$ (avec la distance de Hamming)
- Si $c(P)$ a au moins k coordonnées nulles, alors $P = 0$

$$\delta = n - k + 1$$

Applications

Dans quels domaines utilise-t-on des codes correcteurs ?

Applications

Dans quels domaines utilise-t-on des codes correcteurs ?

- Internet : cadres Ethernet, TCP/IP, IPv4...
- Téléphonie mobile, TNT...
- Communications avec les satellites
- Communications spatiales lointaines
- Stockage de données (depuis les cassettes à bande magnétique)

Exemple : image transmise à la NASA par un satellite en orbite autour de la Lune

