Théorie de l'information

Cours 7 - Introduction au codage canal

Laurent Oudre
laurent.oudre@univ-paris13.fr

Université Paris 13, Institut Galilée
Master Ingénierie et Innovations en Images et Réseaux - 1ère année
2017-2018

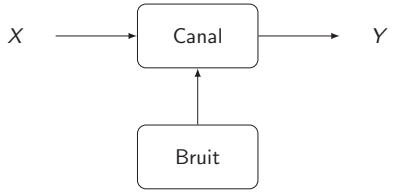
Sommaire

Sommaire

Cadre du cours

- Dans le chapitre précédent, on a représenté chaque symbole (ou groupe de symboles) de la source sous forme d'une série de 0 et de 1.
- En utilisant les propriétés de la source (notamment les probabilités d'apparition des symboles), on a pu faire en sorte de compresser au maximum les données
- Nous allons maintenant étudier la suite de la communication, c'est à dire la transmission à travers le canal.
- On sait que la transmission à travers le canal va être source d'erreurs et de pertes.
- La question que l'on doit se poser est : comment à partir de la sortie d'un canal, retrouver le message que l'on avait envoyé ?

Contexte



- On considère un canal discret sans mémoire.
- On se place après l'étape de codage source, ce qui fait que l'on traite maintenant une nouvelle source X, plus nécessairement sans mémoire, dont l'alphabet est $\mathcal{X} = \{0, 1\}$
- On note Y la variable aléatoire liée à la sortie du canal. Elle prend ses valeurs dans $\mathcal{Y}$ qui n'est pas nécessairement identique à $\mathcal{X}$

Information mutuelle

- Dans un canal non bruité, on a exactement $Y = X$, et les variables X et Y contiennent exactement la même information
- Dans le cas général, on a vu que l'information mutuelle $I(X; Y)$ représentait la quantité d'information commune à X et Y.
- $I(X; Y)$ nous permet donc de quantifier le lien entre l'entrée et la sortie du canal
- Si $I(X; Y)$ est très élevée, cela signifie que Y a beaucoup d'information en commun avec X : il sera donc facile d'estimer X à partir de Y
- En revanche si $I(X; Y)$ est très faible, il n'y a pas ou peu de liens entre X et Y, et on aura du mal à retrouver l'entrée X à partie de la sortie Y

Information mutuelle

- Intuitivement, la quantité $I(X; Y)$ dépend du niveau du bruit, donc des propriétés du canal
- On peut donc se demander : étant donné un canal, quelle est la valeur maximale de $I(X; Y)$, c'est à dire celle permettant d'avoir le lien maximal entre l'entrée et la sortie ?
- On va donc définir une quantité, dépendant uniquement du canal (et pas de l'entrée), représentant l'information mutuelle maximale, c'est à dire le meilleur cas de figure possible.

Capacité d'un canal

Capacité d'un canal

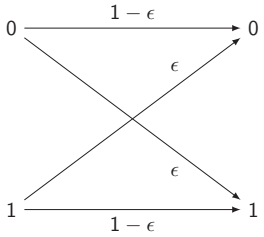
Étant donné un canal discret sans mémoire, ayant pour entrée $X \in \mathcal{X}$, et pour sortie $Y \in \mathcal{Y}$, on appelle **capacité du canal** et on note C la quantité :

$$C = \max_{p_X(x)} I(X; Y)$$

Interprétation

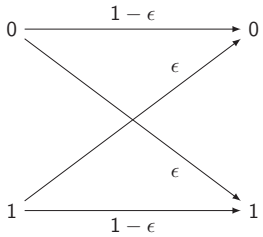
- La capacité d'un canal quantifie le lien maximal possible entre l'entrée et la sortie du canal
- Le terme de capacité fait donc sens, car le canal ne peut pas créer plus de lien entre X et Y, il n'en est pas *capable*
- Nous verrons dans la deuxième partie du cours que grâce au théorème de Shannon, on peut donner une autre interprétation à cette quantité
- Nous allons maintenant voir comment calculer en pratique la capacité d'un canal.

Calcul de la capacité d'un canal binaire symétrique



- On suppose que $p_X(0) = p$ et $p_X(1) = 1 - p$
- On va calculer $I(X;Y)$ et regarder pour quelle valeur de p (donc pour quelle distribution de X) elle est maximale
- La valeur maximale de $I(X;Y)$ sera la capacité du canal

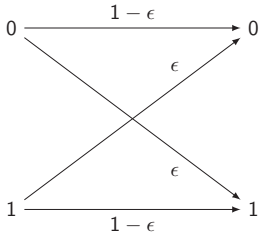
Calcul de la capacité d'un canal binaire symétrique



- $p_Y(0) = p_{Y|X}(0|0)p_X(0) + p_{Y|X}(0|1)p_X(1) = (1-\epsilon)p + \epsilon(1-p) = p + \epsilon - 2p\epsilon$
- $p_Y(1) = 1 - p - \epsilon + 2p\epsilon$

$$H(Y) = -(p+\epsilon-2p\epsilon)\log_2(p+\epsilon-2p\epsilon) - (1-p-\epsilon+2p\epsilon)\log_2(1-p-\epsilon+2p\epsilon)$$

Calcul de la capacité d'un canal binaire symétrique



$$\begin{aligned} H(Y|X) &= -(1-p)\epsilon\log_2(\epsilon) - p\epsilon\log_2(\epsilon) - p(1-\epsilon)\log_2(1-\epsilon) - (1-p)(1-\epsilon)\log_2(1-\epsilon) \\ &= -\epsilon\log_2(\epsilon) - (1-\epsilon)\log_2(1-\epsilon) \end{aligned}$$

- On a :
$$I(X;Y) = H(Y) - H(Y|X)$$
- Donc :
$$I(X;Y) = -(p+\epsilon-2p\epsilon)\log_2(p+\epsilon-2p\epsilon) - (1-p-\epsilon+2p\epsilon)\log_2(1-p-\epsilon+2p\epsilon) + \epsilon\log_2(\epsilon) + (1-\epsilon)\log_2(1-\epsilon)$$

Calcul de la capacité d'un canal binaire symétrique

- Pour calculer le maximum, on annule la dérivée par rapport à p

$$\begin{aligned}
\frac{dI(X;Y)}{dp} = 0 \quad &\iff \quad -(1-2\epsilon)\log_2(p+\epsilon-2p\epsilon) + (1-2\epsilon)\log_2(1-p-\epsilon+2p\epsilon) = 0 \\
&\iff \quad \log_2(p+\epsilon-2p\epsilon) = \log_2(1-p-\epsilon+2p\epsilon) \\
&\iff \quad p+\epsilon-2p\epsilon = 1-p-\epsilon+2p\epsilon \\
&\iff \quad 2p(1-2\epsilon) = 1-2\epsilon \\
&\iff \quad p = \frac{1}{2}
\end{aligned}$$

La valeur de $p = \frac{1}{2}$ est ici logique, car le canal est symétrique

Calcul de la capacité d'un canal binaire symétrique

On a donc :

$$C = 1 + \epsilon\log_2(\epsilon) + (1-\epsilon)\log_2(1-\epsilon)$$

- Si $\epsilon = 0$, il n'y a aucune erreur et $C = 1$ bit
- Si $\epsilon = \frac{1}{2}$, les bits sont indifféremment transmis sous forme de 0 ou de 1. On a donc $C = 0$ bit et les variables d'entrée et de sortie sont indépendantes
- Si $\epsilon = 1$, les 0 deviennent 1 et inversement, on a $C = 1$ bit. Y est exactement l'inverse de X !

Calcul de la capacité d'un canal binaire symétrique

- On aurait pu éviter un long calcul en remarquant que comme $H(Y)$ est le seul terme dépendant de p, il suffit pour que $I(X;Y)$ soit maximale que $H(Y)$ le soit
- Or $H(Y)$ est l'entropie d'une variable aléatoire binaire dans un alphabet $\{0,1\}$ à $M = 2$ éléments, elle est donc maximale pour

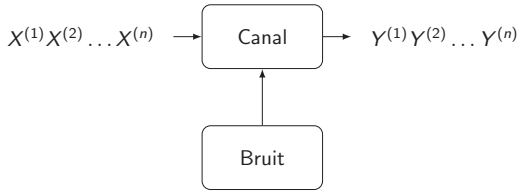
$$p_Y(0) = p_Y(1) = \frac{1}{2} \qquad \text{et vaut dans ce cas } H(Y) = \log_2(2) = 1 \text{ bit}$$

- On a donc

$$I(X;Y) \leq 1 + \epsilon\log_2(\epsilon) + (1-\epsilon)\log_2(1-\epsilon)$$

et l'on retrouve donc naturellement la capacité du canal

Cas d'un message de longueur n



- Nous avons pour le moment considéré l'envoi, la transmission et la réception d'un unique bit
- Nous allons maintenant étudier le cas où l'on transmet un message contenant n bits
- $X^{(t)}$ représente la variable aléatoire liée au symbole émis au temps t
- $Y^{(t)}$ représente la variable aléatoire liée au symbole reçu au temps t

Cas d'un message de longueur n

- On notera pour simplifier les notations :

$$\underline{Y} = Y^{(1)} \dots Y^{(n)} \qquad \underline{X} = X^{(1)} \dots X^{(n)}$$

- Comme le canal est sans mémoire, on a :

$$p_{\underline{Y}|\underline{X}}\left(y^{(1)} \dots y^{(n)} | x^{(1)} \dots x^{(n)}\right) = \prod_{t=1}^{n} p_{Y|X}\left(y^{(t)}|x^{(t)}\right)$$

- On a vu que l'information mutuelle pouvait nous aider à évaluer la proximité de l'entrée et la sortie du canal.
- Que peut-on dire sur $I(\underline{X};\underline{Y})$, l'information contenue à la fois dans $\underline{X}$ et $\underline{Y}$?

Information mutuelle

$$I(\underline{X},\underline{Y}) = H(\underline{Y}) - H(\underline{Y}|\underline{X})$$

$$\begin{aligned}
H(\underline{Y}|\underline{X}) &= -\sum_{\underline{x}\in\mathcal{X}^n}\sum_{\underline{y}\in\mathcal{Y}^n} p_{\underline{X}\underline{Y}}\left(\underline{x},\underline{y}\right)\log_2\left(p_{\underline{Y}|\underline{X}}\left(\underline{y}|\underline{x}\right)\right) \\
&= -\sum_{\underline{x}\in\mathcal{X}^n}\sum_{\underline{y}\in\mathcal{Y}^n} p_{\underline{X}\underline{Y}}\left(\underline{x},\underline{y}\right)\log_2\left(\prod_{t=1}^{n} p_{Y|X}\left(y^{(t)}|x^{(t)}\right)\right) \text{ canal sans mémoire} \\
&= -\sum_{t=1}^{n}\sum_{\underline{x}\in\mathcal{X}^n}\sum_{\underline{y}\in\mathcal{Y}^n} p_{\underline{X}\underline{Y}}\left(\underline{x},\underline{y}\right)\log_2\left(p_{Y|X}\left(y^{(t)}|x^{(t)}\right)\right) \\
&= -\sum_{t=1}^{n}\sum_{x^{(t)}\in\mathcal{X}}\sum_{y^{(t)}\in\mathcal{Y}} p_{XY}\left(x^{(t)},y^{(t)}\right)\log_2\left(p_{Y|X}\left(y^{(t)}|x^{(t)}\right)\right) \\
&= \sum_{t=1}^{n} H(Y^{(t)}|X^{(t)})
\end{aligned}$$

Information mutuelle

$$I(\underline{X};\underline{Y}) = H(\underline{Y}) - H(\underline{Y}|\underline{X})$$

$$\begin{aligned}
H(\underline{Y}) &= H(Y^{(1)} \dots Y^{(n)}) \\
&= H(Y^{(1)}, \dots, Y^{(n)}) \\
&\leq \sum_{t=1}^{n} H(Y^{(t)})
\end{aligned}$$

Information mutuelle

- On a donc :

$$I(\underline{X};\underline{Y}) \leq \sum_{t=1}^{N} H(Y^{(t)}) - \sum_{t=1}^{n} H(Y^{(t)}|X^{(t)})$$

- Donc finalement :

$$I(\underline{X};\underline{Y}) \leq \sum_{t=1}^{n} I(X^{(t)};Y^{(t)})$$

- Or par définition on a :

$$\forall t \qquad I(X^{(t)};Y^{(t)}) \leq C$$

- Donc finalement :

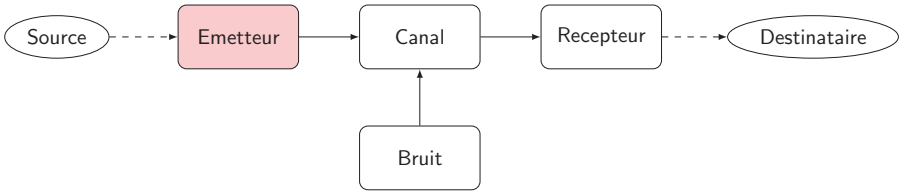
$$I(\underline{X};\underline{Y}) \leq n \times C$$

Cas d'un message de longueur n

- La capacité d'un canal limite donc le lien possible entre le message de sortie et le message d'entrée
- Connaissant cette contrainte, on peut se demander : si la capacité du canal est très faible, pourra-t-on tout de meme retrouver X à partir Y sans faire aucune erreur ?
- Intuitivement, on a envie de répondre non et de dire que s'il y a trop de bruit, la tache sera impossible.
- Shannon a au contraire prouvé que cela était possible, à condition d'introduire avant l'envoi sur le canal une étape supplémentaire, appelée codage canal, qui va rajouter des bits supplémentaires. C'est le deuxième théorème de Shannon.

Sommaire

Émetteur



L'émetteur prend ce message numérique et réalise les étapes suivantes :

- **Codage source :** compression des données pour qu'elles prennent le moins de place possible. Cela revient à remplacer le message à envoyer par un message le plus court possible, souvent représenté sous forme d'une série de 0 et de 1.
- **Codage canal :** rajout de bits d'information supplémentaires dans le message pour permettre de corriger les éventuelles erreurs de transmission
- Transformer le message numérique en un signal physique (onde électromagnétique, signal électrique, etc...) qui puisse être transmis sur le canal de transmission

Travaux de Shannon

La théorie de l'information et les travaux de Shannon ont permis de répondre à deux questions fondamentales sur les sytèmes de communication :

- Codage source : étant donnée une source, à quel point peut-on compresser les données lors du codage, tout en faisant en sorte que le destinataire puisse toujours déchiffrer les messages que l'on envoie ?
  Premier théorème de Shannon
- Codage canal : étant donné un canal de communication bruité, jusqu'à quel débit d'information peut-on envoyer les données en conservant une probabilité d'erreur à la sortie qui soit limitée ?
  Deuxième théorème de Shannon

Principe du codage canal

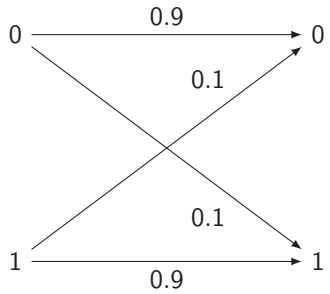
- Après l'étape de codage source, le message à envoyer est représenté sous forme d'une série de 0 et de 1 dont la taille a été optimisée.
- Le canal de transmission étant bruité, le récepteur va recevoir un message où certains bits auront été perdus ou modifiés
- Lorsqu'on décodera le message, il y a donc certains symboles qu'on ne pourra pas retrouver
- Pour éviter ou limiter ces erreurs, on va introduire des bits supplémentaires dont le rôle va être de détecter ou de corriger les éventuelles erreurs de transmission
- Avec le codage source on a essayé de limiter le plus possible le nombre de bits... avec le codage canal on va au contraire en rajouter

Code à répétition

- Pour comprendre le principe du codage canal, nous allons dérouler un exemple simple.
- Imaginons une salle extrêmement bruyante, avec une personne à chaque bout de la salle. Comment faire en sorte que le message transmis de l'une à l'autre soit transmis sans erreur ?
  - Idée 1 : Parler plus fort ou crier. Cela revient à augmenter la puissance d'émission (cf cours de Télécommunications)
  - Idée 2 : Répéter plusieurs fois jusqu'à ce que le message passe. Cela revient à introduire une redondance dans le message envoyé.
- Formalisons maintenant cet exemple par un schéma de communication.

Code à répétition

- Considérons une source, qui après l'étape de codage source, peut être vue comme une variable aléatoire X à valeurs dans $\{0, 1\}$ où les deux symboles sont équiprobables.
- On considère le canal de transmission suivant :



Le bruit est modélisé par une probabilité d'erreur de 0.1

On appellera Y la sortie (ce que recevra le destinataire) à valeurs dans $\{0, 1\}$

Code à répétition

- On suppose que la source répète deux fois chaque bit
  - Lorsque le destinataire reçoit 00, il peut supposer que c'est 0 qui a été envoyé
  - Idem pour 11
  - Mais si le destinataire reçoit 01 ou 10, c'est impossible à trancher
- Une meilleure solution est que la source répète trois fois chaque bit
  - Il suffit dans ce cas de compter le nombre de 0 et de 1 reçus
  - S'il y a plus de 0 que de 1, on suppose que c'est 0 qui a été envoyé
  - Idem pour 1

Code à répétition

Quelle est la probabilité de faire une erreur sur le bit envoyé ?

- Ce sera la même pour 0 et 1 car le canal est symétrique
- Si on a une erreur, c'est par exemple que l'on a plus de 1 que de 0 dans le message reçu, alors qu'on avait envoyé un 0
- Si on suppose que tous les bits envoyés successivement sont indépendants et identiquement distribués, le nombre d'erreur dans un groupe de 3 bits suit une loi binomiale avec $n = 3$ et $p = 0.1$.
- Faire une erreur revient à avoir soit 2, soit 3 bits erronés dans le groupe.

$$P_{err} = C_3^2 p^2 (1-p) + C_3^3 p^3 = 0.028$$

Code à répétition

- En multipliant par 3 le nombre de bits envoyés, on a divisé par 3.6 la probabilité d'erreur ($0.1 \rightarrow 0.028$)
- On peut faire la même chose en considérant 5 répétitions au lieu de 3, et dans ce cas, on a une probabilité d'erreur de 0.0086, ce qui revient à diviser par 11.7 la probabilité d'erreur ($0.1 \rightarrow 0.0086$)
- Le codage canal utilise ce principe : il s'agit d'introduire de la redondance, qui va augmenter la longueur des messages envoyés, mais permettre de diminuer (ou annuler) la probabilité d'erreur.

Quelques notions de codage canal

- Le but de cette partie n'est pas d'étudier en détail les techniques de codage canal, mais de donner quelques définitions et d'introduire quelques concepts permettant de comprendre le principe du codage canal.
- Pour de plus amples informations, on se reportera au cours CDCE optionnel au second semestre
- Le principe du codage canal est toujours le même : on va transformer un message binaire en un autre message binaire de taille plus élevée, afin de le rendre plus robuste aux erreurs.
- Si on code de façon astucieuse, on peut réduire ou annuler la probabilité d'erreur, c'est à dire faire en sorte qu'on puisse retrouver parfaitement ou presque le message émis à partir du message reçu

Codes détecteurs, codes correcteurs

Il existe plusieurs types de codages canal :

- Ceux qui vont introduire de la redondance pour diminuer la probabilité d'erreur (par exemple code à répétition que nous avons traité en exemple)
- Ceux qui vont détecter la présence d'erreurs pour pouvoir éventuellement demander à la source de ré-envoyer le message : codes détecteurs d'erreurs
- Ceux qui vont détecter et corriger les bits erronés : codes correcteurs d'erreurs

Quelques exemples

Quelques exemples

Message à envoyer : 011000

- Code à répétition de longueur 3 : On divise en bloc de taille 1 et on repète chaque bloc 3 fois :

$$011000 \longrightarrow 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0 \longrightarrow 000 \quad 111 \quad 111 \quad 000 \quad 000 \quad 000$$

On a $M = 2$ mots-code possibles : 000 et 111
Ce code nous permet de diminuer la probabilité d'erreur. Pour l'annuler totalement, il faudrait répéter le bit une infinité de fois

- Code de parité de longueur 2 : On divise en bloc de taille 2 et on ajoute un 3 ème bit égal à la somme binaire des deux bits du bloc :

$$011000 \longrightarrow 01 \quad 10 \quad 00 \longrightarrow 011 \quad 101 \quad 000$$

On a $M = 4$ mots-code possibles : 000, 110, 101, 011
Ce code est un détecteur d'erreurs : si on ne reçoit pas l'un de ces 4 mots-codes, on sait qu'il y a eu une erreur sur au moins 1 bit

(M, n)-code : définition

(M, n)-code (HP)

On appelle (M, n)-code un sous ensemble de M mots-code de longueur n réalisés avec l'alphabet $\mathcal{X} = \{0, 1\}$

$$\mathcal{C} = \{\underline{x}_1, \dots, \underline{x}_M\} \subset \mathcal{X}^n$$

- M est le nombre de mots-code constituant le code
- n est la longueur moyenne du code (appelée parfois simplement longueur car tous les mots-code on la même longueur !)
- Le taux (ou rendement) du code est noté R et est défini par :

$$R = \frac{\log_2(M)}{n}$$

Attention ici un mot-code $\underline{x} \in \mathcal{C}$ est un message contenant n bits

(M, n)-code : définition

- On va par exemple diviser le message à envoyer en blocs de k bits, puis chaque bloc de taille k sera codé avec un mot-code de n bits avec $n > k$ pour introduire de la redondance
- Il faut dans ce cas définir M mots-code avec :

$$M = 2^k$$

- Le taux (ou rendement) du code vaut dans ce cas :

$$R = \frac{\log_2(M)}{n} = \frac{k}{n}$$

- R peut donc être interprété comme un **débit d'information**. Sur les n bits envoyés sur le canal, il n'y en a symboliquement que k qui contiennent de l'information : les $n - k$ autres sont juste de la redondance.

(M, n)-code : définition

Code à répétition

Dans notre exemple du code à répétition :

$$011000 \longrightarrow 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0 \longrightarrow 000 \quad 111 \quad 111 \quad 000 \quad 000 \quad 000$$

- On divise en blocs de taille 1, et on affecte à chaque bloc un mot-code de taille 3
- On a considéré 2 mots-code de longueur 3

$$\mathcal{C} = \{000, 111\}$$

- $(2, 3)$-code avec un rendement $R = \frac{1}{3}$

(M, n)-code : règle de décodage

- A la sortie du canal, à cause des erreurs, le récepteur reçoit un message de taille n, mais qui n'est plus nécessairement celui qui a été envoyé.
- Il appartient maintenant à $\mathcal{Y}^n$ et non à $\mathcal{X}^n$! Et ce n'est plus nécessairement un des mots-code de $\mathcal{C}$
- On introduit une **règle de décodage**, associant à chaque élément de $\mathcal{Y}^n$ un mot-code de $\mathcal{C}$

$$\begin{array}{rccc} \Phi : & \mathcal{Y}^n & \mapsto & \mathcal{C} \\ & \underline{y} & \mapsto & \underline{\hat{x}} = \Phi\left(\underline{y}\right) \end{array}$$

(M, n)-code : règle de décodage

Code à répétition

Dans notre exemple du code à répétition :

- On a $\mathcal{Y} = \mathcal{X} = \{0, 1\}$
- Lorsqu'on reçoit un message de 3 bits, on compte le nombre de 0 et de 1.
- Si on a plus de 0 que de 1, on suppose qu'on a envoyé le bit 0, associé au mot-code 000.
- Idem pour 1.
- On a donc :

$$\begin{array}{llll} 000 \rightarrow 000, & 001 \rightarrow 000, & 010 \rightarrow 000, & 100 \rightarrow 000 \\ 110 \rightarrow 111, & 011 \rightarrow 111, & 101 \rightarrow 111, & 111 \rightarrow 111 \end{array}$$

- La règle de décodage est donc :

$$\Phi\left(\underline{y}\right) = \begin{cases} 111 & \text{si nombre de 1 dans } \underline{y} > \text{nombre de 0 dans } \underline{y} \\ 000 & \text{si nombre de 0 dans } \underline{y} > \text{nombre de 1 dans } \underline{y} \end{cases}$$

(M, n)-code : probabilité d'erreur

- On considère un mot-code $\underline{x} \in \mathcal{C}$, qui à la sortie du canal devient $\underline{y} \in \mathcal{Y}^n$. La probabilité d'erreur associée à $\underline{x}$ est la probabilité qu'il y ait une erreur lors de son décodage donc la probabilité :

$$P_{err}\left(\underline{x}\right) = \sum_{\substack{\underline{y} \in \mathcal{Y}^n \\ \Phi\left(\underline{y}\right) \neq \underline{x}}} p_{\underline{Y}|\underline{X}}\left(\underline{y}|\underline{x}\right)$$

- On peut également définir une probabilité d'erreur maximale comme étant :

$$P_{err}^{max} = \max_{1 \leq i \leq M} P_{err}\left(\underline{x}_i\right)$$

(M, n)-code : probabilité d'erreur

Code à répétition

Dans notre exemple du code à répétition :

- $P_{err}(000) = p_{\underline{Y}|\underline{X}}(110|000) + p_{\underline{Y}|\underline{X}}(001|000) + p_{\underline{Y}|\underline{X}}(101|000) + p_{\underline{Y}|\underline{X}}(111|000)$
- Or : $p_{\underline{Y}|\underline{X}}(110|000) = \left(p_{Y|X}(1|0)\right)^2 p_{Y|X}(0|0) = p^2(1-p)$
- Si on suppose que la probabilité d'erreur sur un bit est $p = 0.1$, on a

$$P_{err}(000) = P_{err}(111) = P_{err}^{max} = 0.028$$

Utilisation des (M, n)-codes

- Un choix astucieux d'un (M, n)-code et d'une règle de décodage peut permettre de diminuer ou d'annuler la probabilité d'erreur maximale
- La probabilité d'erreur n'est cependant pas le seul critère lorsque l'on conçoit un codage canal.
- Il faut également faire en sorte que le taux R du code ne soit pas trop faible. Par exemple, si pour annuler la probabilité d'erreur il faut envoyer 10^{10} bits au lieu de 3, cela n'est plus rentable du point de vue télécommunications.
- Le but est donc d'avoir un code avec le taux le plus élevé possible, et permettant tout de même d'annuler la probabilité d'erreur
- Malheureusement, ceci n'est pas toujours possible...

Exemple introductif

- On considère un canal sans mémoire de capacité C, et un (M, n)-code $\mathcal{C}$ de taux R, qui permet d'avoir une probabilité d'erreur maximale nulle.
- Supposons (pour simplifier) que les mots-code $\underline{x}_1, \dots, \underline{x}_M$ sont équiprobables
- On va s'interesser à la transmission d'un mot-code le long du canal
- Notons $\underline{X} \in \mathcal{C}$ la variable aléatoire associée à l'entrée du canal et $\underline{Y} \in \mathcal{Y}^n$ celle associée à la sortie du canal
- Calculons $I(\underline{X}; \underline{Y})$ pour évaluer l'information contenue à la fois dans $\underline{X}$ et dans $\underline{Y}$

Exemple introductif

$$I(\underline{X}; \underline{Y}) = H(\underline{X}) - H(\underline{X}|\underline{Y})$$

- Comme la probabilité d'erreur maximale est nulle, il ne reste aucune incertitude sur $\underline{X}$ lorsque l'on connait $\underline{Y}$:

$$H(\underline{X}|\underline{Y}) = 0$$

- Comme tous les mots-code de $\mathcal{C}$ sont équiprobables, on a :

$$H(\underline{X}) = \log_2(M) = n \times R$$

- D'après ce que nous avons vu au début du cours sur la transmission d'un message de n bits sur un canal discret sans mémoire, on a :

$$I(\underline{X}; \underline{Y}) \leq n \times C$$

- On a donc :

$$R \leq C$$

Exemple introductif

- Dans cet exemple on a vu que si l'on veut annuler la probabilité d'erreur maximale, le taux du code R ne peut pas dépasser C.
- La capacité peut donc s'interpréter comme le taux maximal permettant de définir un code et une règle de décodage annulant la probabilité d'erreur maximale
- Nous avions vu que le taux d'un code pouvait être interprété comme un débit d'information
- On retrouve donc ici un des résultats du deuxième théorème de Shannon : la capacité d'un canal correspond au débit maximal d'information permettant d'annuler la probabilité d'erreur maximale

Deuxième théorème de Shannon

Deuxième théorème de Shannon ou Théorème du codage canal (HP)

- Étant donné un réel $\epsilon > 0$, un canal discret sans mémoire de capacité C et un réel $R < C$, il est possible de construire un code de taux (ou rendement) R et une règle de décodage Φ, tels que :

$$P_{err}^{max} < \epsilon$$

- Corollaire : Étant donné un canal discret sans mémoire de capacité C, il n'existe aucun code de taux (ou rendement) $R > C$ permettant d'avoir

$$P_{err}^{max} = 0$$

Interprétation

- Contrairement au premier théorème, la démonstration de ce théorème (hors programme) ne fournit aucun indice sur comment construire un tel code...
- On découvre ici une deuxième interprétation du terme de capacité : le canal n'est pas capable de transmettre sans erreur avec un taux supérieur à C
- En pratique, pour annuler réellement la probabilité d'erreur, il faut considérer des valeurs extremement élevées de n, qui ne sont donc pas implémentables dans la pratique
- Le deuxième théorème de Shannon décrit donc plutot des propriétés asymptotiques que des résultats utilisables dans la vie réelle
- Néanmoins, il founit des bornes utiles pour évaluer les performances des codes détecteurs et correcteurs d'erreurs.