Introduction à la cryptographie

Sandrine Julia

Université Côte d'Azur

Option – Licence 3 Info./ Math.-Info./ Sc. & Techn. – Automne 2023

1 – Introduction à la théorie de l'information

Plan du cours

https://lms.univ-cotedazur.fr/course/view.php?id=13766

https://upinfo.univ-cotedazur.fr/~julia/Crypto

0 - Introduction au calcul symbolique
1 - Introduction à la théorie de l'information
2 - Codes de source - codes d'Huffman
3 - Codes correcteurs - codes de Hamming
4 - Cryptographie à clef secrète
5 - Compression de texte
6 - Cryptographie à clef publique I
7 - Cryptographie à clef publique II
8 - Fonctions de hachage, intégrité
9 - Signatures digitales
10 - Certificats, authentification
11 - La cryptographie de demain

(Les TP utilisent `Sympy` et la librairie `OpenSSL`)

Système de communication

- la communication fait référence à la circulation d'information entre une source et un récepteur
- la source émet un message qui, avant d'être lu par le récepteur, va transiter dans un canal
- toute pertubation du message due au canal s'appelle le bruit
- la nature de la source peut être variée : on ne considère ici que des sources discrètes ($\neq$ continues) et sans mémoire (*non-markoviennes*)
- on peut ajouter de la redondance au message en cas de bruit
- on peut aussi vouloir garder à tout prix le message secret

Source $\rightarrow$ Codage $\rightarrow$ Canal $\rightarrow$ Décodage $\rightarrow$ Récepteur

Code

- intuitivement, un code est une méthode de transformation qui convertit la représentation d'une information en une autre (cette définition nécessitera d'être précisée selon le contexte)
- selon le but recherché, les codes ont différents usages :
  - efficacité de la transmission : compression des données
  - sécurité de l'information : cryptage, authentification
  - intégrité du message : détection, correction des erreurs
- à l'action de coder (le codage), on couple le décodage qui a pour but de restituer tout ou partie des messages émis par la source
- on distingue le codage avec perte d'information du codage sans perte
- on dissocie le codage de source du codage de canal : le premier est une représentation (binaire) concise d'un message, le second vise à adapter la sortie du premier au canal où elle va transiter.

Exemples

- des codes de source
  - le code Morse
  - le code Braille
  - le code ASCII, ASCII étendu, Unicode ...
- des codes pour compresser
  - RLE (Run Length Encoding)
  - code d'Huffman
  - codage arithmétique
  - (Lempel-Ziv)
- des codes correcteurs d'erreurs
  - codes linéaires
  - codes de Hamming
  - codes de Reed-Solomon
  - codes convolutifs
- des codes pour chiffrer
  - code de Vernam
  - DES, AES
  - RSA, El Gamal

Une très longue histoire

- à Sparte en - 404 av. J-C : la scytale
- à Rome vers - 50 av. J-C : le code de César

$$A \rightsquigarrow D$$
$$B \rightsquigarrow E$$
$$C \rightsquigarrow \ldots$$

- en Europe en 1586 : le chiffre de Vigenère (*cf.* TD nº 1)
- à Paris en 1829 : le code Braille
- avec la télégraphie en 1832 : le code Morse

A	·–	B	–···	C	–·–·	D	–··
E	·	F	··–·	G	– –·	H	····
I	··	J	·– – –	K	–·–	L	·–··
M	– –	N	–·	O	– – –	P	·– –·
Q	– –·–	R	·–·	S	···	T	–
U	··–	V	···–	W	·– –	X	–··–
Y	–·– –	Z	– –··				

- aux USA en 1919, le code de Vernam : $M \oplus K = C$

Théorie de l'information

- la théorie de l'information est due à Claude E. Shannon
  A Mathematical Theory of Communications, 1948
- théorie mathématique basée sur les probabilités et visant à décrire les systèmes de communication
- elle a subi l'influence d'autres théoriciens de l'informatique : A. Turing, J. von Neumann, N. Wiener et présente des convergences avec les travaux de R.A. Fisher
- le problème est celui de la communication entre une source et un récepteur : la source émet un message que le récepteur lit
  on voudrait quantifier l'information que contient chaque message émis

  > *il est clair que si l'émetteur dit toujours la même chose, la quantité d'information apportée par une répétition supplémentaire est nulle*

- l'entropie existe en version combinatoire, en probabilités discrètes ou encore en probabilités continues
- la quantité d'information n'est pas une propriété intrinsèque d'un certain objet, mais une propriété de cet objet en relation avec un ensemble de possibilités auquel il appartient
  dans le cas contraire, on a recours à la complexité de Kolmogorov.

Information combinatoire

- on considère un ensemble de possibilités Ω
- le message consiste à spécifier un élément de Ω
- pour l'instant, on suppose que tous les éléments de Ω sont aussi vraisemblables les uns que les autres
- quelle est l' information transmise par le message : *"Telle possibilité s'est réalisée."* ?
- si Ω a n éléments, on peut spécifier un élément de Ω en donnant $log_2(n)$ informations élémentaires (bits au sens *binary units* ou *Shannon*)

  en numérotant les éléments de Ω
  et en en donnant la décomposition en base 2

- spécifier un élément parmi un ensemble Ω de possibilités revient donc à transmettre $log \mid \Omega \mid$ unités d'information.

  ce sera l'écriture simplifiée de $\log_2(\mid \Omega \mid)$ *dorénavant utilisée*

- on note $I_\Omega(x)$ la quantité d'information de l'événement x appartenant à l'ensemble Ω, on a donc :

$$I_\Omega(x) = log \mid \Omega \mid$$

Information combinatoire (suite)

- petit à petit, on va tendre vers un cadre **probabilisé**
- quelle est l'information de la phrase *"L'événement réalisé appartient à un sous-ensemble A de l'ensemble Ω des possibilités."* ? Intuitivement ...

  si on dit que l'événement réalisé appartient à une partie A de Ω, puis qu'on spécifie ensuite de quel événement de A il s'agit, on a totalement spécifié l'événement, comme si on l'avait donné directement dès le début

- spécifier directement l'événement réalisé, c'est transmettre $log \mid \Omega \mid$ unités d'information
- spécifier un événement, en *sachant déjà qu'il appartient à un sous-ensemble A*, peut se faire en transmettant $log \mid A \mid$ unités d'information
- ainsi, en précisant que l'événement appartient à A, on avait déjà transmis $log \mid \Omega \mid - log \mid A \mid$ unités d'information d'où :

$$I_\Omega(A) = log \mid \Omega \mid / \mid A \mid$$

Un peu de proba. discrètes

- on suppose que toutes les possibilités de Ω ne sont plus forcément équiprobables

  l'idée est que les événements les plus rares sont ceux avec le plus d'information

- on a vu que spécifier l'appartenance d'un élément de Ω à la partie A contient en unités d'information :

$$I_\Omega(A) = log \mid \Omega \mid / \mid A \mid$$

- si les événements de Ω étaient équiprobables, la probabilité de la partie A serait :

$$p(A) = \mid A \mid / \mid \Omega \mid$$

- si (Ω, p) est un espace probabilisé et A est une partie de Ω, l'information apportée par la réalisation d'un événement de A est donc :

$$I_\Omega(A) = log\, 1/p(A) = -\, log\, p(A)$$

Définition de l'entropie

- en particulier, pour un événement x de Ω, on obtient :

$$I_\Omega(x) = log\, 1/p(x)$$

  on remarque que la réalisation d'un événement rare contient plus d'information que celle d'un événement certain qui n'apporte aucune information

- l'entropie est l'information moyenne obtenue en tirant un élément de Ω suivant la probabilité p :

$$\begin{aligned} H(\Omega, p) &= \sum_{x \in \Omega} p(x)\, I_\Omega(x) \\ &= -\sum_{x \in \Omega} p(x)\, log\, p(x) \\ &= \sum_{x \in \Omega} p(x)\, log\, 1/p(x) \end{aligned}$$

l'entropie s'interprète aussi comme le nombre moyen minimal de symboles binaires par lettre nécessaires pour représenter un alphabet.

Entropie : un 1er exemple

1.
   - soit une urne avec 4 boules : $\Omega = \{R, B, J, V\}$
   - on tire une boule **au hasard** et on indique sa couleur
   - l'entropie est ici **maximale** :

$$H(\Omega, p) = 4\ (\frac{1}{4}\ log_2(\frac{1}{1/4})) = log_2(4) = 2$$

   - si on code les couleurs par 00, 01, 10, 11, l'information d'un tirage correspond à 2 bits

Entropie : un 2e exemple

2.
   - si l'urne contient $R_1, R_2, R_3, R_4, B_1, B_2, J$ et V l'entropie devient :

$$H(\Omega, p) = \frac{1}{2}\log_2(2) + \frac{1}{4}\log_2(4) + \frac{1}{8}\log_2(8) + \frac{1}{8}\log_2(8)$$

$$H(\Omega, p) = \frac{7}{4}$$

   - si les codes respectifs sont 0, 10, 110, 111, l'information sur la couleur tirée occupe 1 bit une fois sur deux, 2 bits une fois sur quatre et 3 bits une fois sur quatre, soit en moyenne pondérée : 7/4 bits.

Propriétés de l'entropie

Soit $S = (\Omega, p)$ une source munie d'une distribution de probabilités p :

- on a vu que l'entropie mesure la quantité d'information en moyenne d'une source
- on dit aussi que l'entropie mesure l'incertitude d'une source et en effet elle s'annule si un élément a une probabilité de 1 :

$$0 \leq H(S)$$

- l'entropie est maximale pour une distribution de probabilité uniforme d'où son nom de mesure du désordre, ce qui permet d'obtenir :

$$H(S)\ \leq\ log\ |\ \Omega\ |$$

Ainsi :

$$\boxed{0 \leq H(S)\ \leq\ log\ |\ \Omega\ |}$$

Inégalité de Gibbs

Soit Ω une source finie munie de 2 distributions de probabilité discrètes p et q

Lemme (utilitaire) : l'inégalité suivante est satisfaite :

$$\sum_{x\in\Omega} p(x)\ log\ q(x)/p(x) \leq 0$$

avec égalité si p identique à q.

Preuve (dans le seul cas du logarithme népérien) :

- on a $ln\ r \leq r - 1$ et l'égalité obtenue pour $r = 1$
- on pose $r = q(x)/p(x)$
- on obtient :

$$\sum_{x\in\Omega} p(x)\ ln\ q(x)/p(x) \leq \sum_{x\in\Omega} p(x)\ (\ q(x)/p(x) - 1\ ) = 1 - 1 = 0$$

Majoration de l'entropie

Soit $S = (\Omega, p)$ une source munie d'une distribution de probabilités p :

- on obtient une majoration de l'entropie à partir du lemme de Gibbs
  où la seconde distribution de proba q est choisie uniforme : $\forall x, 1 \leq x \leq |\Omega|$, $q(x) = 1/|\Omega|$

$$\sum_{x\in\Omega} p(x)\ log\ q(x)/p(x) \leq 0$$

$$\sum_{x\in\Omega} p(x)\ (log\ q(x) - log\ p(x)) \leq 0$$

$$\sum_{x\in\Omega} p(x)\ (log\ (1/|\Omega|) - log\ p(x)) \leq 0$$

$$\sum_{x\in\Omega} p(x)\ log\ (1/|\Omega|) + \sum_{x\in\Omega} p(x)\ log\ 1/p(x) \leq 0$$

- or $H(S) = \sum_{x\in\Omega} p(x)\ log\ 1/p(x)$ donc :

$$H(S) \leq -\sum_{x\in\Omega} p(x)\ log\ (1/|\Omega|)$$

$$H(S) \leq -\sum_{x\in\Omega} p(x)\ (log\ 1 - log\ |\Omega|)$$

$$H(S) \leq \sum_{x\in\Omega} p(x)\ log\ |\Omega|$$

$$H(S) \leq log\ |\Omega|$$

En Python

```
from math import *

s = {'a': 0.1,'b': 0.1,'c': 0.25,'d': 0.15,'e': 0.35,'f': 0.05}


def entropie (source) :

    res = 0
    for ch in source :
        p = source[ch]
        res = res + ( p * (log(1/p,2)))

    return res

print(entropie(s))       #  -->  2.321127473337187
```

(*cf.* TP n° 1)

A suivre ...

Une notion fondamentale

- le calcul de l'entropie d'une source de messages donne une mesure de l'information minimale que l'on doit conserver afin de représenter ces données sans perte
- l'entropie permet d'estimer la qualité des codes utilisés en vue du chiffrement
- elle permet une **minoration** de la longueur moyenne des mots d'un code : c'est le (premier) Théorème de Shannon.
- en compression de fichiers informatique, l'entropie indique le nombre minimal de bits que peut atteindre un fichier compressé

  l'entropie d'une source qui émet un caractère 1 avec une probabilité 0,1 et le caractère 0 avec une probabilité 0,9 est de 0,14 contre 0,3 dans le cas de probabilités uniformes
  ainsi, une faible entropie permet d'envisager une bonne compression ...

- en pratique, l'entropie de l'image ou du son est abaissée en retirant des détails imperceptibles pour les humains, comme lors de la compression des sons par le format MP3, des images par JPEG ou des vidéos par MPEG.