Concepts mathématiques

Théorie de l'information

Théorie de Shannon

- Mesure la confidentialité d'un code à partir de
  - L'incertitude sur le texte clair
  - La connaissance du texte crypté
- Confidentialité parfaite :
  - La connaissance du texte crypté ne dévoile rien sur le texte clair.
- La quantité d'information d'un message :

  = le nombre minimal de bits nécessaires pour coder toutes les significations possibles de ce message

Entropie

- Mesure l'incertitude sur l'information et dénotée H(M)
- Définit le nombre de bits d'information que l'on doit connaître pour reconstituer un message crypté
- En général, H(M) est de type $\log_2 n$ (n = nombre de significations possibles)

Cryptographie -

Entropie ≙ nombre de bits de texte clair qui doivent être retrouvés pour retrouver le texte en clair en entier

La variable aléatoire X renseigne totalement sur le déroulement de l'expérience E si H(X)=H(E).

Exemple :

Jours de la semaine : 3 bits d'information

000 = dimanche, 001 = lundi, …, 110 = samedi

Genre : 1 bit d'information

1 = masculin, 0 = féminin

Or en Ascii, beaucoup plus de place mémoire mais pas plus d'information

Exemple

- On lance un dé non pipé et on considère les 3 variables aléatoires suivantes :
  - X_1 qui vaut 0 si le nombre tiré est pair, 1 s'il est impair.
  - X_2 qui vaut 0 si le nombre tiré est 1 ou 2, 1 si le nombre tiré est 3 ou 4, 2 si le nombre tiré est 5 ou 6.
  - X_3 qui vaut le nombre tiré.
- Il est intuitivement clair que la connaissance de X_3 renseigne plus sur le déroulement de l'épreuve aléatoire que la connaissance de X_2, qui elle-même renseigne plus que celle de X_1
- On a $H(X_1) = \log 2 < H(X_2) = \log 3 < H(X_3) = \log 6$

Cryptographie -

Autre exemple :

On vous présente 10 cartons sur lesquels sont inscrits sur la face cachée un nombre (tous les nombres sont différents).

Vous pouvez poser des questions du type : "Est-ce que le nombre sur ce carton est plus élevé que sur celui-là".

Vous payez un franc par question, et vous en recevez 15 lorsque vous savez réordonner les cartons.

Acceptez-vous de jouer?

Il y a 10! façons d'ordonner les cartons. L'incertitude sur l'ordre dans lequel ils sont rangés vaut donc log(10!)=21,8 bits environ. Chaque réponse apporte au plus un bit d'information. Il faudra, du point de vue de la théorie de l'information, 22 questions en moyenne pour reconstituer l'ordre. Il ne faut pas jouer!

Entropie - approche ponctuelle

- h(x) = log[1/p(x)] = - log p(x)
- Un événement certain apporte une information nulle (h(1) = 0)
- Si x et y sont 2 événements indépendants :
  - p(x,y) = p(x) .p(y)
  - h(x,y) = log[1/p(x,y)] =- log p(x,y)

    ou = log[1/ p(x).p(y)] = log[1/p(x)] + log[1/p(y)]

    = h(x) + h(y)

Remarque : travail avec des minuscules pour distinguer les approches

h(x) ne dépend pas de la valeur de x mais seulement de la probabilité qui lui est associée

Entropie - approche ponctuelle

- Quantité d'information associée à x conditionnelle à la réalisation de y :
  - h(x | y) =- log p(x | y)
  - Par Bayes : p(x, y) = p(x|y).p(y) = p(y|x).p(x)
  - Donc :
    - h(x,y) = - log(p(x,y)) = - log(p(x|y).p(y)) = -log(p(x,y)) – log(p(y))
    - = h(x|y) + h(y) = h(y|x) + h(x)
- Information mutuelle :
  - i(x ; y) = h(x) – h(x|y) = log(p(x|y) / p(x)) = i(y ; x)

Information mutuelle : quantité d'information que la connaissance d'une des variables apporte sur l'autre

Entropie – approche générale

- Soient 2 v.a. X à valeurs dans $(x_1,\ldots,x_n)$ et Y dans $(y_1,\ldots,y_n)$
- $H(X) = \sum_i p(x_i) \log(1/p(x_i))$
- Entropie conjointe :
  - $H(X,Y) = -\sum_i \sum_j p(x_i,y_j) \log(p(x_i|y_j))$
  - Or
    - $P(x_i|y_j) = p(x_i,y_j)/p(y_j)$ et $p(y_j) = \sum_i p(x_i,y_j)$
  - Donc
    - $H(X,Y) = \sum_i \sum_j p(x_i,y_j) \log(1/ p(x_i|y_j).p(y_j))$
    - Et comme log(a.b) = log(a) + log(b)

Remarque :

•On suppose les choix successifs des x_i indépendants

•Le remplacement d'un symbole par un autre ne change pas la valeur de H (probabilités équidistribuées)

Entropie – approche générale

- On obtient
  - $H(X,Y) = \sum_i \sum_j p(x_i, y_j) \log(1/p(x_i|y_j)) + \sum_i\sum_j p(x_i,y_j) \log(1/p(y_j))$
  - $H(X,Y) = H(X|Y) + H(Y) = H(Y|X) + H(X)$
- $H(X|Y) \triangleq$ incertitude qu'il reste sur X lorsque Y est connu

$\rightarrow$ l'information apportée par les 2 variables X et Y vaut l'information apportée par Y seule plus l'information apportée par X connaissant déjà la valeur de Y

- Information mutuelle moyenne
  - $I(X;Y) = \sum_i\sum_j p(x_i,y_j)\log(p(x_i|y_j)/p(x_i))$
  - $I(X;Y) = H(X) – H(X|Y) = H(Y) – H(Y|X)$
  - $I(X;Y) = H(X) + H(Y) – H(X,Y)$

Remarque

•H(X) \geq 0

•Le maximum de H(X) est atteint pour n fixé, lorsque p_i = 1/n, \forall i

Taux du langage

- Taux du langage (r)
  - r = H(M)/N
    - N = la longueur des messages
    - Taux anglais : plusieurs valeurs entre1 et 1,5 bit par lettre pour un grand N. (habituellement H(M) = 1,3 bits par lettre)
- Taux absolu (R)
  - Nombre maximal de bits pouvant être codés par chaque caractères (séquences de caractères équiprobables)
  - $R = \log_2 L$
    - L = nombre de caractères dans un langage
    - R = entropie maximale des caractères isolés

r = nombre moyen de bits d'information par caractère

Redondance

- En anglais, $R = \log_2 26 = 4,7$ bits par lettre
- Langue fortement redondante
- La redondance d'un langage (D) :
  - D = R – r
  - Exple en anglais :
    - D = 4,7 – 1,3 = 3,4 bits par lettre d'information redondante
    - Message ascii : D = 8 – 1,2 = 6,8 bits d'information redondante

D mesure la redondance du langage utilisé pour exprimer le texte clair en bits par lettre

Différence entre le nombre de bits maximal qui peuvent être codés par caractères et du nombre réellement nécessaire de bits pour obtenir une information sur le caractère

Sécurité d'un cryptosystème

- Hyp : Cryptanalystes disposent d'informations probabilistes concernant M (langage et redondance associée)
- But : par analyse, modifier les probabilités associées aux textes clairs possibles

⇒ un texte émerge parmi les possibilités

- Confidentialité parfaite = le texte chiffré ne fourni aucune information concernant le texte clair :
  - Shannon → uniquement si nombre de clés possibles aussi grand que nombre de messages possibles
  - En pratique → Seul le *masque jetable* correspond car *longueur clé* = *longueur texte* et *clé non réutilisée*

Le but des cryptanalystes est de déterminer la clef K, le texte clair M ou les deux. Toutefois, ils peuvent se contenter de certaines informations probabilistes concernant M si c'est du son, un texte francais, des données pour un tableur, etc.

Pour la plupart, les cryptanalystes ont quelques informations probabilistes concernant M avant de commencer. Ils connaissent probablement le langage du texte en clair. Ce langage a une certaine redondance associée. Si c'est un message destiné à Bob, il commence probablement par « cher Bob ». Le but de la cryptanalyse est, par analyse, de modifier les probabilités associées avec tous les textes en clair possibles. Finalement, un texte clair considéré comme certain émergera du paquet des textes clairs possibles.

Exemple : le chiffre de Vernam

- Soient :
  - une phrase de m lettres.
  - n = ensemble des messages possibles = 26^m (= ensemble des clés et des chiffrés possibles)
  - Clés équiprobables
- On a $P(K = k) = \frac{1}{n} = \frac{1}{26^m}$
- Et donc : $P(M = x|C = y) = P(M = x)$
  - x et y respectivement tout texte clair et tout texte chiffré

Démonstration (1)

On cherche à prouver la formule :

$$P(M = x|C = y) = P(M = x).$$

Par définition, on a :

$$P(M = x|C = y) = \frac{P(M = x, C = y)}{P(C = y)}.$$

Orn on sait que C,K et M sont reliés par la relation C=M+K (l'addition se faisant modulo 26). On a donc :

$$\begin{aligned} P(M = x, C = y) &= P(M = x, K = y - x) \\ &= P(M = x)P(K = y - x) \\ &= \frac{P(M=x)}{26^m} \end{aligned}$$

Démonstration (2)

où on a utilisé que M et K sont indépendants. D'autre part :

$$\begin{aligned} P(C = y) &= \sum_z P(C = y|M = z)P(M = z) \\ &= \sum_z P(K = z - y)P(M = z) \\ &= \sum_z \frac{P(M=z)}{26^m} \\ &= \frac{1}{26^m}. \end{aligned}$$

Ceci achève de prouver la formule.

Confidentialité parfaite

- Par un petit calcul rapide, on en déduit que :

$$H(M|C) = H(M)$$

- Le chiffre de Vernam est un système cryptographique parfait.
- **Théorème :**

  Si un système cryptographique est parfait, alors :

$$H(K) \geq H(M)$$

H(M|C) = H(M) ⇔ P(M|C) = P(M) ⇔ P(C|M) = P(C) ⇔ I(C;M) = 0

I(C;M) signifie que M et C sont indépendants

H(K) ≥ H(M) : Dans un cryptosystème parfait, il y a au moins autant d'incertitude sur les clés que sur les messages.

Conséquence, dans un système sûr, les clés sont aussi volumineuses que les messages

Démonstration (1)

Preuve du théorème : D'après les propriétés de l'entropie conditionnelle, pour n'importe quel système cryptographique :

$$\begin{aligned} H(M|C) &\leq H\big((M,K)|C\big) \\ &\leq H(K|C) + H\big(M|(C,K)\big). \end{aligned}$$

Maintenant, puisqu'un système de cryptographie est réversible, la connaissance de C et K détermine complètement M. Par conséquent :

$$H\big(M|(C,K)\big) = 0.$$

On a donc :

$$H(M|C) \leq H(K|C) \leq H(K).$$

Si le système cryptographique est parfait :

$$H(M) \leq H(K).$$

Sécurité d'un cryptosystème

- Inévitablement, C donne des informations sur M
- Objectif : minimiser l'information fournie
- Remarques :
  - Plus un texte est redondant, plus il est simple à cryptanalyser
  - la compression réduit la redondance
  - H(K) = $\log_2$(nombre de clés)
  - Clés de 64 bits = entropie de 64
  - Plus l'entropie est grande et plus il est difficile de casser le cryptosystème

Un système parfait est différent d'un système pratique : la clé doit être petite et réutilisable

Distance d'unicité

- Nombre de clés $\neq$ traduisant un msg chiffré en texte clair intelligible : $2^{H(K) - nD} - 1$
- Distance d'unicité (U) = longueur de texte chiffré minimale requise pour pouvoir s'attendre à ce qu'il n'y ait qu'un seul déchiffrement sensé (mesure probabiliste) : $U = H(K)/D$
- Plus grande est U et meilleur est le cryptosystème
  - DES + clé 56 bits : 8,2 caractères ASCII, 66 bits
- U est inversement proportionnel à D

Répond à la question : *à partir de quand notre système n'est-il plus sûr ?*

Des textes chiffrés significativement plus courts que cette distance ont des chances d'avoir plusieurs déchiffrement également valables et donc augmentent la sécurité, car il est difficile à l'attaquant de choisir le bon déchiffrement

La distance d'unicité indique le minimum de texte chiffré pour lequel il est probable qu'il n'y ait qu'un seul texte clair plausible correspondant quand une attaque exhaustive est montée.

La distance d'unicité n'est pas une mesure de la quantité de texte chiffré qu'il faut pour la cryptanalyse mais bien de la quantité de texte chiffré nécessaire pour qu'il n'y ait qu'une solution raisonnable à la cryptanalyse.

Un cryptosystème peut être inviolable par calcul même si théoriquement il est possible de le casser avec une faible quantité de texte chiffré.

Quand la redondance approche de zéro, même un chiffrement trivial peut être inviolable par une attaque à texte chiffré seulement.

Démonstration (1/)

- Soient
  - M = $M_1 \ldots M_n$
  - K une permutation de l'alphabet $\rightarrow$ H(K) = log(26!)
  - C = $C_1 \ldots C_n$
- On définit la distance d'unicité comme le plus petit entier d tel que : $H(K|(C_1, ..., C_d)) = 0$
- D'autre part, on a :

$$H(K|(C_1, ..., C_d)) = H(K, C_1, ..., C_d) - H(C_1, ..., C_d)$$
$$= H(K, M_1, ..., M_d) - H(C_1, ..., C_d)$$

Optons pour la modélisation suivante : le message est constitué d'une succession de lettres $M_1 \ldots M_n$, la clé K est une permutation de l'alphabet à 26 lettres. En particulier, H(K)=log(26!). Le chiffré est une succession de lettres $C_1 \ldots C_n$.

Concrètement, d désigne le nombre moyen de lettres du message chiffré qu'il faut connaître pour pouvoir déterminer la clé.

Démonstration (2/)

- D'où : $H(K|(C_1, ..., C_d)) = H(K) + H(M_1, ..., M_d) - H(C_1, ..., C_d)$

  Cette égalité provient du fait que la clé est indépendante du message. (H(M,K) = H(K) + H(M))
- On a donc : $H(K) + H(M_1, ..., M_d) - H(C_1, ..., C_d) = 0$
- Or :
  - H(K) = log(26 !)
  - Par la définition de la redondance d'un langage, on constate que : $H(M_1 \ldots M_d) \simeq r$ et $H(C_1 \ldots C_d) \simeq R$
- On en déduit que U = d = H(K)/D = H(K)/ R-r

D'autre part, on a : $H(K|(C_1, ..., C_d)) = H(K, C_1, ..., C_d) - H(C_1, ..., C_d)$
$= H(K, M_1, ..., M_d) - H(C_1, ..., C_d)$

(connaître la clé et le message chiffré équivaut à connaître la clé et le message en clair).

D'où : $H(K|(C_1, ..., C_d)) = H(K) + H(M_1, ..., M_d) - H(C_1, ..., C_d)$

Cette égalité provient du fait que la clé est indépendante du message. On a donc :

$$H(K) + H(M_1, ..., M_d) - H(C_1, ..., C_d) = 0$$

Exemple : Calcul de l'entropie d'un texte

- Comment estimer l'entropie d'un texte M écrit en français et constitué de d lettres $M_1,...,M_d$?
- Si les lettres pouvaient être considérées comme indépendantes les unes des autres, on aurait $H(M_1...M_d) = d\ H(M_1)$.
- Mais ce n'est pas le cas! Après un L, il est beaucoup plus probable que l'on ait un A ou un E qu'un R.

Exemple : Calcul de l'entropie d'un texte

- En revanche, au-delà de 3 ou 4 lettres d'écart, il y a beaucoup moins d'interactions : si on a un L en 1ère position, il n'est pas très clair que l'on aura en 5è position plutôt un A ou plutôt un R.
- Cela signifie que la suite

$$H(M_1), \frac{H(M_1M_2)}{2}, \frac{H(M_1M_2M_3)}{3}, \ldots$$

va rapidement s'approcher d'une valeur limite H.

Exemple : Calcul de l'entropie d'un texte

- Pour d suffisamment grand (en pratique, d>20 suffit), on pourra faire l'approximation :

$$H(M_1 \ldots M_d) = dH$$

- Le calcul effectué sur un texte français de grande taille a donné comme estimation pour H : H $\simeq$ 3,97.
- Remarquons que la valeur de H dépend fortement de la langue employée. Un calcul du même type sur un texte anglais (ou allemand) a donné une valeur de H sensiblement plus grande (de l'ordre de 4,2).

Exemple : distance d'unicité

- Texte francais :
  - Par analyse statistique sur de nombreux textes francais : r $\simeq$ 3,97
  - De nouveau, par analyse statistique : R $\simeq$ 4,67
- On en déduit que

$$U \simeq \frac{H(K)}{R-r} = \frac{\log(26!)}{4,67-3,97} \simeq 126$$

- En pratique, il faut effectivement de l'ordre de ce nombre de lettres pour retrouver la clé

Confidentialité idéale

- Confidentialité idéale = système où la distance d'unicité est infinie
- Un cryptosystème idéal n'est pas forcément parfait (réciproque toujours vraie)
- Si un cryptosystème offre une confidentialité idéale, même une cryptanalyse réussie laissera une incertitude quant à savoir si le texte décodé est le bon texte

En pratique

- La cryptanalyse réelle se base rarement sur ces concepts
  - Peu d'algorithmes pratiques sont totalement imperméables à l'analyse
- Cependant, concepts parfois utiles.
  - P.ex. déterminer l'intervalle de changement de clefs pour un algorithme déterminé
- Cryptanalystes : tests statistiques basés sur la théorie de l'information pour mener des analyses dans les directions les plus efficaces

Confusion et diffusion

- Techniques de base pour gommer les redondances dans un texte clair
- Confusion :
  - gomme les relations entre texte clair et texte chiffré.
  - Moyen : substitution → remplacement
- Diffusion :
  - Disperse la redondance du texte clair en la répartissant dans le texte chiffré
  - Moyen : transposition – permutation → réarrangement

Concepts mathématiques

Théorie de la complexité

Théorie de la complexité

- Fournit une méthodologie pour analyser la complexité de calcul de différents algorithmes et techniques cryptographiques
- Compare les algorithmes et les techniques cryptographiques pour déterminer leur niveau de sécurité
- Remarque :
  - Théorie de l'information : tous les algos peuvent être cassés
  - Théorie de la complexité : nous apprend s'ils peuvent etre cassés avant la fin du monde…

Complexité des algorithmes

- Déterminée par la puissance de calcul nécessaire pour l'exécuter
- Complexité calculatoire est mesurée par 2 paramètres :
  - T (complexité en temps) – S (complexité en espace)
  - T et S sont exprimés en fonction n, n = taille de l'entrée
  - Donne un ordre de grandeur (terme de la fonction qui croit le plus vite)

Complexité d'algorithme

- Si un algorithme a une complexité $O(n^t)$ – appelé polynomial
- Si un algorithme a une complexité $O(t^{f(n)})$ – appelé exponentiel
- Idéalement : tout algorithme de « cassage » dot être exponentiels en temps.

Classe	Complexité	Nombre d'ops pour $n=10^6$	Temps pour 10^6 ops/sec
Quadratiques	$O(n^2)$	10^{12}	11,6 jours
Exponentiels	$O(2^n)$	$10^{301\,030}$	$10^{301\,006}$ fois l'age de l'univers

Hypothèse : unité de temps de l'ordinateur est la microseconde

Complexité de problèmes

- détermine les temps et espaces minimaux nécessaires pour résoudre l'instance la plus difficile du problème sur un ordinateur théorique (machine de Turing)
- Problèmes résolvables :
  - En temps polynomial → solubles
  - En temps exponentiel → non solubles
- Classes :
  - P : problèmes résolus en temps polynomial
  - NP : problèmes résolus en temps polynomial sur une MTND

La théorie de la complexité classe aussi la complexité inhérente des problèmes et pas seulement la complexité d'algorithmes particuliers utilisés pour résoudre des problèmes.

La théorie détermine les temps et espaces minimaux nécessaires pour résoudre l'instance la plus difficile du problème sur un ordinateur théorique connu sous le nom de machine de Turing. Il s'agit d'une machine à états finis avec une mémoire en écriture et en lecture sous la forme d'un ruban infini. Cette machine est un modèle réaliste du calcul.

Les problèmes qui peuvent être résolus avec des algorithmes polynomiaux en temps sont appelés solubles, car ils peuvent généralement être résolus en un temps raisonnable pour des entrées de taille raisonnable.

Les problèmes qui ne peuvent être résolus en temps polynomial sont appelés non solubles car calculer leur solution devient vite impossible. Il s'agit souvent des problèmes résolus par des algorithmes exponentiels.

Turing a également prouvé que certains problèmes étaient indécidables. Il est impossible de concevoir un algorithme pour les résoudres et certainement pas un algorithme polynomial.

MTND (machine de Turing non déterministe) : variante de la machine de Turing normale qui devine les solutions. La machine devine une solution d'un problème – soit en faisant par chance une bonne hypothèse, soit en essayant toues les possibilités en parallèle – et vérifie son hypothèse en temps polynomial.

Problème NP-Complet

- Probleme NP-Complet :
  - SAT : *étant donné une formule booléenne propositionnelle, y a-t-il un moyen d'assigner des valeurs de vérité aux variables de telle manière que la formule soit vraie ?*
  - Si SAT peut être résolu en temps polynomial alors on a NP = P.
  - Inversement, si on peut prouver que tout problème dans NP n'a pas un algorithme déterministe polynomial en temps, cela démontrerait que SAT n'admet pas non plus d'algorithme déterministe polynomial en temps.
  - Problèmes aussi difficile que tout autre problème dans NP

Problème NP-Complet

- Tout problème aussi difficile que SAT est dit NP-COMPLET également.
  - Voyageur de commerce, groupement par 3, 3 SAT,…
- Difficulté : rien ne prouve que NP = P
- Or de nombreux problèmes sont trivialement cassables en temps polynomial non déterministe et si NP = P, ils seraient cassables par des algorithmes déterministes utilisables !
- Objectifs :
  - Méthode de chiffrement/déchiffrement $\in$ P
  - Méthode de décryptement $\in$ NP $\rightarrow$ NP-Complet

Questions

- Expliquer
  - Entropie – entropie conjointe – entropie conditionnelle
  - Information mutuelle
  - Taux du langage – redondance
  - Confidentialité parfaite – distance d'unicité
  - Confusion – diffusion
  - Complexité d'algorithme – de problèmes
  - Classes de problèmes

Références

- Stinson – chapitre 2
- Schneier – chapitre 11
- http://www.bibmath.net/crypto/complements/entropie.php3
- http://www-igm.univ-mlv.fr/~beal/Enseignement/TheorieInfo/
- http://www.univ-mlv.fr/enseignement/cours/informatique/laurent/nouvbd/chap52.htm