Complexité de Kolmogorov et notion de mot aléatoire

Aurel PAGE

20 décembre 2007

Introduction

Kolmogorov a fondé la théorie actuelle des probabilités, basée sur la théorie de la mesure. Bien que cette théorie soit toujours en vigueur et utilisée par tous les mathématiciens probabilistes, Kolmogorov n'en était pas satisfait, en particulier parce qu'elle ne nous apprend rien sur ce qu'est un nombre aléatoire. C'est une approche totalement différente de l'aléatoire qu'il a ensuite envisagée, basée sur des notions d'informatique théorique, à laquelle on s'intéressera dans ce document. Le but est de pouvoir remplacer un énoncé tel que "si $X_1, X_2, \dots$ est une suite de variables aléatoires de loi uniforme dans $\{0,1\}$, alors presque sûrement on a telle propriété" par "si $x_1x_2\dots$ est un mot aléatoire sur $\{0,1\}$, alors on a telle propriété."

Conventions

Dans tout le document, on se place sur l'alphabet $A = \{0,1\}$ et toutes les machines de Turing considérées auront A pour alphabet d'entrée, auront une bande de travail avec un même alphabet B, et une bande de sortie avec pour alphabet A. $|x|$ désigne la longueur du mot x.
Quand on parlera d'ordre lexicographique sur les mots, on entend par là que si deux mots ont des longueurs différentes, alors le plus court est le plus petit, et s'ils sont de même longueur on applique l'ordre lexicographique usuel. On peut ainsi enumérer (de façon calculable) tous les mots finis dans l'ordre lexicographique. On identifiera un entier et sa représentation en binaire.

1 Définition, propriétés élémentaires

1.1 Définition, optimalité

La complexité de Kolmogorov mesure la quantité d'information contenue dans un mot, et le définit comme la taille du plus petit programme qui engendre ce mot. On notera qu'elle ne fait pas intervenir le temps nécessaire à la production du mot, et que la machine considérée peut ne pas s'arrêter.

Définition 1.1. Soit M une machine de Turing. Pour $x \in A^*$, on définit la complexité de x relativement à M ainsi :

$$K_M(x) = \min\{|y||M(y) = x\}$$

On posera $\min \emptyset = +\infty$.
On peut l'interpréter ainsi : M est un algorithme de décompression, y est une version compressée de x.
Lorsqu'il existe, on notera $d_M(x)$, qu'on appellera description minimale de x, un mot tel que $|d_M(x)| = K_M(x)$ et $M(d_M(x)) = x$. La fonction partielle ainsi définie d_M est injective : si $d_M(x) = d_M(x')$ alors $M(d_M(x)) = x$ et $M(d_M(x')) = x'$ donc $x = x'$.

Exemple 1.1. Si M est un extracteur de fichiers .zip, la complexité d'un mot est alors la taille de sa version compressée en .zip.

La définition qu'on a donnée dépend de la machine choisie. Cependant, le théorème suivant assure qu'il est possible de choisir une machine de sorte que la complexité soit définie de façon universelle à une constante près.

Définition 1.2. On dit qu'une machine M est asymptotiquement meilleure qu'une machine M' si :

$$\exists C \in \mathbb{N}\ \forall x \in A^*\ K_M(x) \leq K_{M'}(x) + C$$

On dit qu'une machine est optimale si elle est asymptotiquement meilleure que toute machine.

Remarque 1.2. La constante dépend de l'autre machine.

Exemple 1.3. Si M est une machine capable de décompresser le ZIP et le TAR, et M' capable seulement capable de décompresser le ZIP, M est asymptotiquement meilleure que M'.

Théorème 1.4 (d'invariance)**.** *Il existe une machine optimale.*

L'idée est celle de l'auto-extraction. M sera maintenant un interpréteur d'un langage de programmation, et y la description de la méthode de décompression et du mot, ainsi que sa version compressée.

Démonstration. On considère une machine universelle M, alors cette machine est optimale.
En effet, soit M' une autre machine. Soit $C = |< M', \epsilon >|$.
Alors si $M'(y) = x$, $M(< M', y >) = x$ et $|< M', y >| = C + |y|$ donc $K_M(x) \leq K_{M'}(x) + C$. □

Exemple 1.5. Si M est une machine qui prend en entrée un programme écrit en Caml et l'exécute, et $x = 01^{1000000}$, alors on peut prendre pour y :
`let rec f=0->()|n->print_string"01";f(n-1)in f(1000000);;`
Ce programme fait 57 caractères, donc si on le code en ASCII, on obtient $|y| = 456$, d'où $K_M(x) \leq 456$.
On remarque que ceci est beaucoup moins que si on avait donné un code ne contenant qu'une instruction consistant à écrire le mot en entier, qui aurait demandé au moins 2000000 bits : on a compressé le mot x. x est peu aléatoire en ce sens qu'il contient peu d'information par rapport à sa taille : il est très répétitif.

Dans la suite, on fixe une machine optimale M et on note $K(x) = K_M(x)$ et $d(x) = d_M(x)$.

1.2 propriétés élémentaires

Proposition 1.6. *On a les propriétés suivantes :*

1. $\forall x \in A^* \ K(x) \leq |x| + O(1)$
2. *Si f est une fonction calculable, alors* $K(f(x)) \leq K(x) + O(1)$.
3. $\forall n \ \text{card}\{x | K(x) \leq n\} < 2^{n+1}$

Démonstration.

1. Soit I une machine qui recopie l'entrée sur la sortie : $\forall x \in A^* \ I(x) = x$ et $K_I(x) = |x|$. Alors M est asymptotiquement meilleure que I.
2. Soit N une machine telle que : $\forall y \in A^* \ N(y) = f(M(y))$. M est asymptotiquement meilleure que N, donc : $\forall x \in A^* \ K(f(x)) \leq K_N(f(x)) + O(1)$. Or $\{y | N(y) = f(x)\} \supset \{y | M(y) = x\}$ donc $\forall x \in A^* \ K_N(f(x)) \leq K(x)$, d'où le résultat.
3. Soit n entier. On note $B_n = \{x | K(x) \leq n\}$. Alors $d(B_n) \subset (A + \epsilon)^n$, d est injective et $\text{card}(A + \epsilon)^n = 2^{n+1} - 1$.

□

Remarque 1.7. La propriété 2. formalise le fait qu'une fonction calculable ne peut pas créer d'information.

Exemple 1.8. Pour n entier, la fonction $f : x \longmapsto x^n$ (au sens de la concaténation) est calculable.
Donc $K(x^n) \leq K(x) + O(1)$: répéter le même mot plusieurs fois n'ajoute pas d'information.

1.3 Calculabilité

Théorème 1.9. *K n'est pas calculable.*

Démonstration. Par l'absurde, supposons K calculable.
On définit alors la fonction f ainsi : $f(m)$ est le plus petit mot x (pour l'ordre lexicographique) tel que $K(x) \geq m$. f est bien définie, car il existe des mots de complexité aussi grande qu'on veut d'après la propriété 3. précédente, et est calculable car K l'est.
Alors pour tout m entier, $K(f(m)) \geq m$ par définition, mais f est calculable, donc $K(f(m)) \leq K(m) + O(1)$. De plus $K(m) \leq |m| + O(1)$.
On a donc $m \leq K(f(m)) \leq K(m) + O(1) \leq |m| + O(1)$, ce qui est faux.

□

Théorème 1.10. *K est approchable supérieurement au sens suivant :*

$$K(x) = \lim_{n \infty} \mathrm{k}(x, n)$$

pour tout x tel que $K(x) < \infty$,
où k *est calculable et décroissante en n.*

Démonstration. Soit c tel que $\forall x\ K(x) \leq |x| + c$.
Soit $\mathrm{k}(x,n)$ la taille du plus petit mot de taille inférieure à $|x| + c$ tel que M s'arrête en moins de n étapes de calcul avec x en sortie, s'il existe, et $\mathrm{k}(x,n) = |x| + c + 1$ sinon.
Alors k est calculable, décroissante en n et si $K(x) < \infty$, il existe N entier, tel que M lancée sur $d(x)$ s'arrête en moins de N étapes de calcul avec x en sortie, donc pour tout $n \geq N$, $\mathrm{k}(x,n) = K(x)$. □

2 Notion d'aléatoire pour les mots finis

2.1 Incompressibilité

Définition 2.1. Soit $c \in \mathbb{N}$. Un mot x est dit c-compressible si $K(x) < |x| - c$. Dans le cas contraire x est dit c-incompressible.

Remarque 2.1. A cause des théorèmes 1.9 et 1.10, il est facile d'exhiber des mots compressibles, mais difficile d'exhiber des mots incompressibles.

Proposition 2.2. *Pour tout n et pour tout $c \leq n$, il existe des mots c-incompressibles de longueur n. La proportion de c-incompressibles parmi tous les mots de longueur n est au moins $1 - 2^{-c}$.*

Démonstration. Pour tout n entier et $c \leq n$, on a vu qu'il existe au plus 2^{n-c} mots de complexité inférieure à $n-c-1$, or il existe 2^n mots de longueur n, donc il existe des c-incompressibles de longueur n et la proportion de c-compressibles est au plus $2^{n-c}/2^n = 2^{-c}$. □

Remarque 2.3. Beaucoup de mots sont donc incompressibles : il existe c tel que 99% des mots soient c-incompressibles.

2.2 Caractère aléatoire

On aimerait que ces mots soient aléatoires, au sens où ils sont "typiques" : ils vérifient toutes les propriétés qu'on peut attendre d'un mot pris au hasard. Le théorème qui suit justifie cela.

Définition 2.2. Soit P une propriété sur les mots finis.
On dit que P est vraie pour presque tous les mots si :

$$\lim_{n\infty} \frac{\mathrm{card}\{x | \neg P(x) \wedge |x| \leq n\}}{2^n} = 0$$

Théorème 2.4. *Soit P une propriété calculable, vraie pour presque tous les mots.*
Alors pour tout $c > 0$, il n'y a qu'un nombre fini de c-incompressibles qui ne vérifient pas P.

Démonstration. Soit P une propriété calculable, vraie pour presque tous les mots, $c > 0$.
Soit f la fonction qui à un entier i associe le i^{eme} mot (dans l'ordre lexicographique) qui ne vérifie pas P.
Cette fonction est calculable car P l'est.

On a donc $K(f(i)) \leq K(i) + O(1)$, on note C la constante associée.
Soit C' telle que $\forall i\ K(i) \leq |i| + C'$.
Soit $N \in \mathbb{N}$ tel que

$$\forall n \geq N\ \frac{\text{card}\{x | \neg P(x) \wedge |x| \leq n\}}{2^n} \leq 2^{-C-C'-c}$$

(P vraie pour presque tous les mots).
Soit x qui ne vérifie pas P, tel que $|x| \geq N$. Soit $n = |x|$.
Alors $\exists i \in \mathbb{N}\ f(i) = x$ et on a alors $i \leq 2^{n-C-C'-c}$ donc $K(i) \leq |i| + C' \leq n - C - c$.
Mais alors $K(x) = K(f(i)) \leq K(i) + C \leq n - C - c + C = n - c$ donc x est c-compressible.
Tout mot qui ne vérifie pas P de longueur supérieure à N est c-compressible, donc il n'y a qu'un nombre fini de c-incompressibles qui ne vérifient pas P. □

Exemple 2.5. Soit P la propriété "ne pas être l'écriture en binaire d'un nombre premier".
P est vraie pour presque tout mot : les nombres représentés par un mot de longueur au plus n sont les $k < 2^n$. Or d'après le théorème des nombres premiers, le nombre de nombres premiers inférieurs à N, $\pi(N)$, vérifie $\pi(N) = O(N/\log N)$.
Donc la proportion de mots de taille au plus n représentant un nombre premier est en $O(1/n)$, donc tend vers 0.
Pour tout c, il n'existe donc qu'un nombre fini de c-incompressibles qui représentent un nombre premier.

3 Notion d'aléatoire pour les mots infinis

Rappels :
- $A^{\mathbb{N}}$ est muni d'une topologie produit métrisable (A étant muni de la topologie discrète) et les ouverts sont de la forme $LA^{\mathbb{N}}$ où $L \subset A^*$.
- On peut définir une mesure μ sur $\mathcal{B}(A^{\mathbb{N}})$ telle que $\mu(LA^{\mathbb{N}}) = \sum_{x \in L} 2^{-|x|}$. C'est une mesure de probabilité. ($\mu(A^{\mathbb{N}}) = \mu(\{\epsilon\}A^{\mathbb{N}}) = 2^{-0} = 1$)
- $C \subset A^{\mathbb{N}}$ est négligeable ssi il existe $U_1, U_2, \ldots U_n$ ouverts tels que : $C \subset \bigcap_{n \in \mathbb{N}} U_n$ et $\lim_{n\infty} \mu(U_n) = 0$.

Ceci est à comparer à la topologie et la mesure de Lebesgue sur $[0, 1]$. En effet, le but recherché est de définir la notion de réel aléatoire. Pour un informaticien, un réel est une suite de 0 et de 1 (son développement en binaire).

3.1 Première tentative de définition

On va tenter une première approche, par le bas (en s'intéressant à chaque mot individuellement), en s'inspirant de ce qu'on a fait pour les mots finis.

Définition 3.1. Soit $x \in A^{\mathbb{N}}$. Pour n entier, on note $x|n = x_1 x_2 \ldots x_n$.
On dit que x est fortement aléatoire si $K(x|n) \geq n + O(1)$.

Le problème... c'est que c'est un échec !

Théorème 3.1 (des grandes oscillations)**.** *Il n'existe pas de mot fortement aléatoire.*
Plus précisément, pour tout mot infini x, il existe une infinité de n tels que $K(x|n) \leq n - \log n + O(1)$.

L'idée de la preuve est la suivante : dans un mot x, il y a toujours un peu plus d'information que x lui-même : il y a en plus $|x|$, qui est une information de taille $\log|x|$. Voici comment on l'utilise :

Démonstration. Tout d'abord, on a :

$$\log n - \log(n - \lfloor \log n \rfloor) = \log(1 - \frac{\lfloor \log n \rfloor}{n}) = 1 - \frac{\lfloor \log n \rfloor}{n} + o(\frac{\log n}{n})$$

donc $\log n - \log(n - \lfloor \log n \rfloor) \leq 1$ pour n assez grand.
Soit x un mot infini, m un entier, $u = x|m$.
$1u$ est l'écriture en binaire d'un entier N.
Soit v tel que $uv = x|N$.
$m = |u| = \lfloor \log n \rfloor \in \{\log(N - \lfloor \log N \rfloor), \log(N - \lfloor \log N \rfloor) + 1\}$
$= \{\log|v|, \log(|v|) + 1\}$
Soit D la machine qui prend en entrée bv, où $b = \log(N - \lfloor \log N \rfloor) - m$ (1 bit).
D calcule alors $N - m = |v|$, puis calcule m avec b et l'encadrement ci-dessus, puis calcule $N = N - m + m$. Alors D a calculé $1u = N$, donc u. Elle écrit alors uv sur la sortie.
$K_D(uv) \leq |bv| = N - \lfloor \log N \rfloor + 1$.
M est asymptotiquement meilleure que D, donc il existe c telle que $K(x) \leq K_D(x) + c$. On a donc $K(x|N) = K(uv) \leq K_D(uv) + c \leq N - \lfloor \log N \rfloor + 1 + c = N - \log N + O(1)$, et N prend une infinité de valeurs. □

Remarque 3.2. En faisant une preuve plus générale, on peut remplacer log par n'importe quelle $f : \mathbb{N} \longrightarrow \mathbb{N}$ calculable telle que $\sum_{n\in\mathbb{N}} 2^{-f(n)} = \infty$.

3.2 Autre approche

On va donc tenter une autre approche, par le haut (en s'intéressant aux mots aléatoire dans leur ensemble). Une idée naïve pourrait être de dire qu'un mot est aléatoire s'il n'est contenu dans aucun ensemble de mesure nulle, mais ceci est voué à l'échec car tout mot est contenu dans le singleton le contenant, qui est de mesure nulle. On va donc ajouter quelques contraintes supplémentaires.

Définition 3.2. On dit que $X \subset A^{\mathbb{N}}$ est constructivement de mesure nulle si il existe des ouverts $U_n = X_n A^{\mathbb{N}}$ tels que :
- $X \subset \bigcap_{n\in\mathbb{N}} U_n$
- $\{(n, u)|u \in X_n\}$ est récursivement énumérable.
- $\mu(U_n) \leq 2^{-n}$

$\bigcap_{n\in\mathbb{N}} U_n$ est qualifié de G_δ constructivement de mesure nulle.
On appelle énumérateur de recouvrement pour X tout énumérateur de $\{(n, u)|u \in X_n\}$.

Remarque 3.3. Un ensemble constructivement de mesure nulle est négligeable.

Exemple 3.4. Si $x = x_1x_2\ldots$ et si $n \longmapsto x_n$ est calculable, alors $\{x\}$ est un G_δ constructivement de mesure nulle.
(Il suffit de prendre $X_n = \{x|n\}$ et d'appliquer la définition.)
En particulier, le développement binaire d'un rationnel, d'un algébrique, de π, de e, sont contenus dans des ensembles constructivement de mesure nulle.

Lemme 3.5. *Soit $Y_0, Y_1, Y_2, \ldots$ des ensembles constructivement de mesure nulle tels qu'il existe une fonction calculable qui à i associe un énumérateur de recouvrement pour Y_i.*
Alors $Y = \bigcup_{n\in\mathbb{N}} Y_i$ est constructivement de mesure nulle.

Démonstration. Il s'agit de trouver un énumérateur de recouvrement pour Y. Ceci est équivalent à trouver une fonction calculable qui à n associe un énumérateur de X_n tel que, en notant $U_n = X_n A^{\mathbb{N}}$, $Y \subset U_n$ et $\mu(U_n) \leq 2^{-n}$.
En effet il suffit d'énumérer successivement les n premiers éléments des n premiers X_i pour avoir un énumérateur de recouvrement.
Réciproquement si on a un énumérateur de recouvrement, le transformer en énumérateur qui ne sélectionne que les couples (k, u) où $k = n$ est calculable : lorsque la machine initiale écrit sur la bande de sortie, on la fait passer par un état intermédiaire qui vérifie si $k = n$ et qui envoie, si la condition n'est pas vérifiée, sur une copie de la machine qui n'écrit plus.
On utilise alors l'énumérateur suivant E_Y :

```
Entrée : n
1     Pour i de 1 à ∞
2         Pour j de 1 à i
3             Calculer le (j+n+1)-ème énumérateur E de X_j.
4             Imprimer les i premiers éléments énumérés par E.
```

E_Y énumère, sur une entrée n, une réunion, notée Z_n, de X_i tels que $Y_i \subset X_i A^{\mathbb{N}}$ et $\mu(X_i A^{\mathbb{N}}) \leq 2^{-i-n-1}$ d'où $Y \subset Z_n A^{\mathbb{N}}$ et $\mu(Z_n A^{\mathbb{N}}) \leq 2^{-n}$. □

Proposition 3.6. *Il y a un plus grand ensemble constructivement de mesure nulle. On le notera G.*

Remarque 3.7. Ceci est faux pour les négligeables : les singletons sont négligeables mais leur réunion est $A^{\mathbb{N}}$ qui est de mesure 1.

Démonstration. La proposition est équivalente à $G = \bigcup_{g\in\mathcal{G}} g$ est constructivement de mesure nulle, où $\mathcal{G}$ est l'ensemble des G_δ constructivement de mesure nulle.
On va appliquer le lemme de la façon suivante : on construit une fonction calculable qui à tout entier k associe un énumérateur de recouvrement, telle que tout énumérateur de recouvrement soit atteint pour un k. On aura alors le résultat.
On procède ainsi : l'ensemble des énumérateurs de couples (n, u) peut être énuméré ; on les note $E_1, E_2, \ldots$.
La fonction qu'on construit prend k en argument. Elle trouve E_k. Elle le modifie ensuite de sorte à ce que si c'est un énumérateur de recouvrement, il reste inchangé, et sinon il soit remplacé par un énumérateur de recouvrement, ce qui donnera la propriété recherchée : E_k est remplacé par E'_k qui énumère les mêmes éléments que E_k mais retient, pour chaque n tel qu'un (n, u) a déjà été énuméré, la mesure de la partie de X_n qui a déjà été énumérée (c'est un nombre rationnel, donc on peut le mettre en mémoire). Si en rajoutant (n, u) on devrait obtenir une mesure strictement plus grande que 2^{-n}, l'énumérateur n'imprime pas (n, u) et continue. Comme cette condition n'arrive jamais pour un énumérateur de recouvrement, on a bien la propriété voulue. □

Définition 3.3. On dit qu'un mot infini x est aléatoire au sens de Martin-Löf si $x \notin G$, ce qui est équivalent à x n'appartient à aucun ensemble constructivement de mesure nulle. On note R l'ensemble des aléatoires au sens de Martin-Löf.

Remarque 3.8. Cette fois-ci, on est sûr que de tels mots aléatoires existent, il y en a même une infinité car $\mu(R) = 1$.

Pourquoi une telle définition ?
Martin-Löf a observé les preuves de la théorie des probabilités, et a remarqué le fait suivant : lorsqu'on prouve une propriété vraie presque sûrement (par exemple la loi des grands nombres), on le fait en montrant que les éléments qui ne vérifient par la loi sont contenus, pour tout k, dans un ouvert de mesure inférieure à 2^{-k}, et que les ouverts construits par ce procédé sont énumérables (au sens défini précédemment).
Donc les mots infinis aléatoires au sens de Martin-Löf vérifient toutes les propriétés de ce type !
Par exemple, tout mot infini $x_1x_2\ldots$ aléatoire au sens de Martin-Löf vérifie :

$$\lim_{n\infty} \frac{x_1 + x_2 + \cdots + x_n}{n} = \frac{1}{2}$$

3.3 Lien avec la complexité de Kolmogorov

Toute tentative de d'approche par le bas, en utilisant la complexité de Kolmogorov, est-elle donc vaine ? Heureusement non, c'est ce qu'établit ce théorème (admis) :

Théorème 3.9 (Miller et Lyu, 2004)**.** *Soit x un mot infini. Alors*
x est aléatoire au sens de Martin-Löf
ssi pour toute fonction calculable $f : \mathbb{N} \longrightarrow \mathbb{N}$ telle que $\sum_{n\in\mathbb{N}} 2^{-f(n)} < \infty$,

$$\exists c\ \forall n\ K(x|n) \geq n - f(n) - c$$

Un grand merci à M. Serge Grigorieff pour m'avoir reçu aussi vite, pour son aide et ses explications passionnantes !

Références

[1] Introduction to the theory of computation, M. Sipser.

[2] Formalisation de la notion de réel aléatoire, S. Grigorieff, LIAFA, CNRS & Paris 7.

[3] Kolmogorov complexity, A. Shen, Uppsala University.