Complexité de Kolmogorov

ENS de Paris — Langages formels, calculabilité et complexité

Pierre Cagne

Résumé

La notion de complexité se rapporte généralement à celle de temps et d'espace dans le domaine de l'algorithmique. Ce document s'intéresse à une autre sorte de complexité : quelle quantité d'information un objet (dans un formalisme choisi) contient-il ?

Cette complexité est nommée *complexité de* Kolmogorov[1]. Le présent document traite de sa définition, de certaines de ses propriétés et d'applications variées de celles-ci.

Table des matières

[1] du nom du mathématicien russe Andreï Kolmogorov

1 Présentation informelle

Intuitivement, il est raisonnable de dire que la chaîne $c =$ "01010101010101" est plus simple que $c' =$ "00111010011000", dans le sens où elle contient moins d'informations. En effet, en français, en cherchant à former des phrases les plus courtes possibles, c est décrite par « "01" concaténé 7 fois », tandis que c' est décrite par « Deux '0', puis trois '1', puis un '0', puis un '1', puis deux '0', puis deux '1', puis trois '0' ».

Ces deux chaînes de caractères nous apparaissent donc clairement ne pas avoir la même *simplicité* bien qu'elles comportent toutes deux 14 caractères (et donc, *a priori*, occupent un même espace).

L'exemple suivant est encore plus frappant :

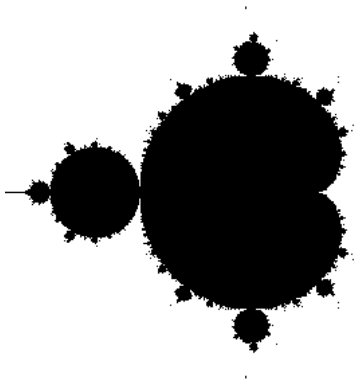

FIG. 1 – Cette image est fortement compressible.

A priori, la figure 1, en tant qu'image de 200 pixels par 200 pixels, contient 40000 *unités d'information* (20Ko sur mon disque dur). Mais bien évidemment, cette image représente l'ensemble de MANDELBROT et est donc descriptible par une implémentation dans un langage quelconque (fichier de 4Ko pour une implémentation OCaml).

Il s'agit maintenant de formaliser l'intuition que peuvent donner de tels exemples.

2 Complexité de Kolmogorov

2.1 Notations et conventions

Dans la suite de ce document, les objets manipulés seront des chaînes de caractères binaires. Ceci n'est en rien une restriction : en effet, tout objet peut être encodé en binaire. Nous travaillons donc désormais sur l'alphabet $\{0, 1\}$.

Étant donné une machine de TURING $\mathcal{M}$, et un mot w, on dit que $(\mathcal{M}, w)$ décrit x si $\mathcal{M}$, avec l'entrée w, s'arrête sur le mot x.

Puisque l'on veut ne travailler qu'avec des données binaires, il est nécessaire d'encoder $(\mathcal{M}, w)$. On procédera de la manière suivante : si $\langle\mathcal{M}\rangle$ est l'encodage binaire[2] de $\mathcal{M}$, on note $\langle\langle\mathcal{M}\rangle\rangle$ le mot obtenu en doublant chaque bit de $\langle\mathcal{M}\rangle$. Alors l'encodage de la description $(\mathcal{M}, w)$ est $\langle\mathcal{M}, w\rangle = \langle\langle\mathcal{M}\rangle\rangle \cdot 01 \cdot w$.

Enfin, la fonction $|\cdot|$ est la fonction longueur sur les mot de $\{0,1\}^*$.

2.2 Définition

Définition. *Soit x un mot sur $\{0,1\}$. La complexité de* KOLMOGOROV *de x, $K(x)$, est défini comme suit :*

$$K(x) = \min\{|\langle\mathcal{M}, w\rangle| | \mathcal{M} \text{ machine de TURING}, \ w \in \{0,1\}^*\}$$

Informellement, $K(x)$ est la description minimale dont nous avons parlé en section 1, *i.e.* la quantité d'information contenue dans x.

2.3 Propriétés immédiates

La propriété suivante décrit le fait que la description d'un objet est nécessairement plus restreinte (au sens large) que l'objet lui-même.

Proposition 1. $\exists c \in \mathbb{R}, \ \forall x \in \{0,1\}^*, \ K(x) \leq |x| + c$

Démonstration. L'idée est que dans le cas contraire, la minimalité de $K(x)$ est contredite par $|x|$ plus court. L'ajout de c est du à la présence d'une machine de TURING dans la description [3] de x.

Rédigeons-le. Soit x un mot binaire. On note alors $\mathcal{M}$ la machine de TURING identité ($\mathcal{M}$ s'arrête pour tout mot en entrée et donne ce même mot en sortie). Alors $(\mathcal{M}, x)$ est une description évidente de x. Si $c = |\langle\langle\mathcal{M}\rangle\rangle| + 2$, alors $|\langle\mathcal{M}, x\rangle| = |x| + c$. Par définition de minimalité de $K(x)$, on a l'inégalité cherchée.

♣

La propriété suivante indique que la description d'une répétition n'est pas une répétition de la description.

Proposition 2. $\exists c \in \mathbb{R}, \ \forall x \in \{0,1\}^*, \ K(xx) \leq K(x) + c$

Démonstration. Soit x un mot binaire. Soit alors $(\mathcal{M}, w)$ une description de x telle que $K(x) = |\langle\mathcal{M}, w\rangle|$.

On définit également $\mathcal{N}$ une machine de TURING qui pour une entrée $\langle\mathcal{P}, v\rangle$ où $(\mathcal{P}, v)$ décrit un mot s, s'arrête et renvoie ss en sortie.[4]

[2] On ne détaille pas ici l'encodage des machines de TURING. Se référer à [Car08] par exemple.

[3] Attention ! Ne pas comprendre que c dépend de la description. Voir démonstration

[4] La construction d'un telle machine est semblable à celle d'une machine de TURING universelle. Doubler la sortie ne change fondamentalement rien à cette construction.

Alors $(\mathcal{N}, \langle \mathcal{M}, w \rangle)$ est alors une description de xx de longueur $|\langle \mathcal{N}, \langle \mathcal{M}, w \rangle \rangle| = K(x) + c$ où $c = |\langle \langle \mathcal{N} \rangle \rangle| + 2$.

♣

2.4 Optimalité de la description

On peut légitimement se demander si la description via les machines de TURING est optimale, ou si un autre langage de description pourrait assurer une meilleure compressibilité.

Le théorème suivant répond à cette interrogation. On appellera désormais langage de description toute fonction $p : \{0,1\}^* \to \{0,1\}^*$ calculable. (Tel qu'un langage de programmation par exemple.) On note K_p la complexité de KOLMOGOROV associé à ce langage (*i.e* $K_p(x) = \min\{|s| | s \in \{0,1\}^*,\ p(s) = x\}$).

Théorème 3. *Soit p langage de description.* $\exists c_p \in \mathbb{R},\ \forall x \in \{0,1\}^*,\ K(x) \leq K_p(x) + c_p$

Démonstration. Tout vient du fait que p est calculable. En effet, il existe $\mathcal{M}_p$, une machine de TURING, qui pour $s \in \{0,1\}^*$ en entrée, s'arrête sur le mot $p(s)$.

Soit donc x un mot binaire. Soit s un mot binaire tel que $K_p(x) = |s|$ et $p(s) = x$ (s est donc une description de s relativement à p). Alors $(\mathcal{M}_p, s)$ est une description de x. Et si $c_p = |\langle \langle M \rangle \rangle| + 2$, on a $\langle \mathcal{M}_p, s \rangle = K_p(x) + c$. On a finalement, par minimalité de $K(x)$: $K(x) \leq K_p(x) + c_p$.

♣

3 Applications de la complexité de Kolmogorov

3.1 K n'est pas calculable

On a vu que K fournit une information quant à la compressibilité d'un chaîne binaire. Cet un outil qui pourrait donc s'avérer primordial quant au stockage de données. Si ceci n'est pas le cas, c'est que K n'est pas calculable. C'est ce que ce paragraphe-ci montre.

Commençons par un lemme qui nous servira ensuite. Il est très similaire à la propriété de doublage[5].

Lemme 4. *Pour toute fonction calculable f, il existe $c \in \mathbb{R}$ tel que* $\forall x \in \{0,1\}^*,\ K(f(x)) \leq K(x) + c$.

Démonstration. La preuve est similaire à celle du doublage. Seule la sortie de la machine de TURING $\mathcal{N}$ de la preuve en question change. ♣

[5] En fait, la propriété de doublage est un cas particulier de ce lemme

Théorème 5. *K n'est pas calculable.*

Démonstration. Afin de prouver cela, nous allons formaliser le paradoxe bien connu « du plus petit entier naturel qui ne peut être décrit en français en moins de vingt mots ».

Supposons de K soit calculable. Alors la fonction f, définie sur $\mathbb{N}$ par $f(n) =$ « premier mot x dans l'ordre lexicographique tel que $K(x) > n$ », l'est également. (Cette fonction est bien définie car le nombre de mot dont la description est de longueur inférieure ou égale à n est au plus de 2^n.)

On a tout d'abord, par définition, $K(f(n)) > n$ pour tout $n \in \mathbb{N}$. On a aussi $c \in \mathbb{R}$ tel que $K(f(n)) \leq K(n) + c$ par le lemme 4. De plus, la propriété 1 donne un $c' \in \mathbb{R}$ tel que $K(n) \leq |n| + c'$.

Comme $|n| = \lfloor \log_2(n) \rfloor$ (n est confondu avec sa représentation binaire), on a finalement une constante $d \in \mathbb{R}$ telle que $\forall n \in \mathbb{N},\ n < \log_2(n) + d$. Ce qui est absurde.

♣

3.2 Théorème d'incomplétude de Gödel

Pour juger de la pertinence d'une notion, il convient d'en observer les applications externes à la notion elle-même. Cette partie montre comment la complexité de Kolmogorov peut permettre de démontrer le théorème d'icomplétude de Gödel.

Bien que nous ne le montrerons pas ici de façon formelle, on peut se convaincre assez aisément que la complexité de Kolmogorov est exprimable dans le langage formel de l'arithmétique. Nous allons donc nous servir du lemme suivant.

Lemme 6. *Il existe une constante $c \in \mathbb{R}$ telle que tous les théorèmes de la théorie du type "$K(x) > n$" satisfassent $n < c_G$.*

Démonstration. Par l'absurde, on suppose notre lemme faux. On va alors construire un algorithme (*i.e.* une fonction calculable) ne respectant pas le lemme 4.

Le voici : pour $k \in \mathbb{N}$ donné, énumerer tous les théorèmes de la théorie et dès qu'un théorème du type $K(x) > s$ avec x quelconque et $s \geq k$ est trouvé, renvoyer x. Nommons cet algorithme α (avec nos notations, on a $x = \alpha(k)$). Étant donné notre hypothèse de raisonnement par l'absurde, $\alpha(k)$ est défini pour tout k.

Par définition même de α, on a $K(\alpha(k)) > k\ \forall k \in \mathbb{N}$. D'autre part, α est calculable, donc par le lemme 4, on a également $\forall k \in \mathbb{N},\ K(\alpha(k)) \leq K(x) + c$ ($c \in \mathbb{R}$ constante). Par le même enchainement d'inégalité qu'en preuve du théorème 5, on aboutit à la contradiction $k < log_2(k) + O(1)$.

♣

Le théorème d'incomplétude de GÖDEL est alors une conséquence immédiate de ce lemme. En effet, il suffit de montrer qu'il existe un mot de complexité de KOLMOGOROV supérieure au c trouvé par le lemme. Alors, "$K(x) > \lfloor c_G \rfloor + 1$" est vrai dans la théorie sans être prouvable par celle-ci (c'est le théorème d'incomplétude).

L'existence d'un tel mot est évidente : le nombre de mot dont la complexité de KOLMOGOROV est inférieure à c_G est de 2^{c_G} au plus.

Cet exemple montre la puissance de la notion de complexité de KOLMOGOROV. La partie suivante propose une application tout aussi étonnante : une modélisation non probabiliste du hasard.

4 Incompressibilité et hasard

Reprenons les deux mots binaires utilisés en introduction comme heuristique de la complexité de KOLMOGOROV : $c =$ "01010101010101" et $c' =$ "00111010011000". En tirant des bits au hasard (avec le jeu de *pile ou face* par exemple), il y a intuitivement peu de chance de tomber sur c (dont la complexité de KOLMOGOROV est faible), alors qu'il n'est pas choquant de tomber sur c' (dont la complexité de KOLMOGOROV est de l'ordre de la longueur de c') après 14 lancés. Aussi est-il légitime de faire le parallèle : les mots de *forte* complexité de KOLMOGOROV modélisent-ils les mots tirés au hasard ?

4.1 Notion d'incompressibilité

La propriété 1 exprime le fait que la description d'un mot est de longueur moindre que lui-même (sinon, autant l'écrire lui-même). La définition et les propriétés suivantes montrent qu'il existe en effet des mots que l'on est obligé d'écrire complètement pour les décrire.

Définition. *On dit qu'un mot x est c-incompressible ($c \in \mathbb{N}$) si $K(x) \geq |x| - c$. Un mot* 0*-incompressible est simplement appelé* incompressible.

Cette définition n'a d'utilité que si elle définit effectivement certains mots. La propriété suivante l'affirme.

Proposition 7. *Pour tout $n \in \mathbb{N}^*$, pour tout $c \in [|0, n|]$, il existe un mot c-incompressible de longueur n.*

Démonstration. Vu ce que l'on a remarqué en début de cette section 4, il y a peu de chance que cette preuve soit constructive. En effet, celle-ci se base simplement sur le nombre de mots respectant une propriété.

Le nombre de mots de longueur n est bien entendu de 2^n. Le nombre de mots de longueur strictement inférieure à $n - c$ est de :

$$\sum_{0 \leq i \leq n-c-1} 2^i = 2^{n-c} - 1$$

$2^n - 2^{n-c} + 1 > 0$ conclut.

♣

Notamment, pour $c = 0$, cette propriété montre qu'il existe des mots x tels que $K(x) = |x|$.

4.2 Infinité des nombres premiers

Aussi surprenant que cela puisse être, la notion d'incompressibilité permet de montrer qu'il y a une infinité de nombres premiers en utilisant une seule et unique propriété arithmétique, la décomposition en facteurs premiers.

Supposons par l'absurde qu'il n'y ait qu'un nombre fini de nombres premiers que l'on note $p_1, p_2, \ldots, p_n$. Alors, pour tout $m \in \mathbb{N}^*$, on a

$$m = p_1^{\alpha_1} p_2^{\alpha_2} \ldots p_n^{\alpha_n}$$

Ainsi, chaque $m \in \mathbb{N}^*$ est décrit par n entiers α_i, $1 \leq i \leq n$. De plus, les exposants sont majorables en fonction de m : en effet, on a la majoration (très large mais suffisante) $\forall i \in [|1, n|],\ \alpha_i \leq \log(m)$. Aussi, chaque entier m peut être décrit par une chaîne binaire de longueur $n \log(\log(m))$. m étant elle-même une chaîne de longueur $\log(m)$, ceci imposerait la non existence de chaîne incompressible, ultimement. On a contradiction avec le lemme 7.

4.3 Modélisation non probabiliste du hasard

Cette partie exprime en quoi notre intuition au sujet du parallèle « hasard/grande complexité » est ou n'est pas fondée. Les mots incompressibles ressemblent en effet aux mots tirés au hasard en cela qu'ils respectent les mêmes propriétés qu'eux.

La première des propriétés que vérifient les mots tirés au hasard est celle d'avoir approximativement le même nombre de 0 et de 1 dans sa composition.

Proposition 8. *Soit* $b : \left\{ \begin{array}{ccc} \{0,1\}^* & \longrightarrow & \mathbb{N} \\ x & \longrightarrow & ||x|_0 - |x|_1| \end{array} \right.$ *. b est majorée sur les mots incompressibles.*[6]

Idée de la preuve. On peut prouver, de façon analogue à la preuve de l'incalculabilité de K, que toute fonction calculable majorée par K est bornée. Comme des mots incompressibles de toute longueur existent, on peut restreindre le théorème aux fonctions calculables majorées par K sur les incompressibles.

C'est le cas de b, calculable, et majorée par $|x| = K(x)$ sur les incompressibles.

♣

[6] $|x|_a$ est le nombre d'occurences de a.

On peut encore prouver que la longueur de la plus grandes succesions de 0 dans un x incompressible est en $O(\log|x|)$, ce qui rapproche encore les mots incompressibles et les mots tirés au hasard.

Enfin, les mots incompressibles vérifient une propriété plus large des mots tirés au hasard. Nous avons tout d'abord besoin de définir la notion de « vraie presque partout ».

Définition. *Une propriété P est une fonction $P : \{0,1\}^* \to \{\texttt{true}, \texttt{false}\}$.*

Définition. *Une propriété P est dite* vraie presque partout *si la proportion des mots de longueur n ne la satisfaisant pas tend vers 0 quand $n \to +\infty$.*

Plus formellement, P est dite vraie pesque partout *si*

$$\frac{\text{Card}\left(\{x \in \{0,1\}^* || x| = n \text{ et } P(x) = \texttt{false}\}\right)}{2^n} \underset{n\to+\infty}{\longrightarrow} 0$$

Le théorème suivant montre que les mots incompressibles sont représentatifs des mots en général, de même que les mots aléatoires.

Théorème 9. *Soit P une propriété* calculable *vraie presque partout. Alors, pour tout $b \geq 0$, il n'y a qu'un nombre fini de mots b-incompressibles ne satisfaisant pas P.*

Démonstration. On construit la machine de Turing $\mathcal{M}$prenant une entrée $i \in \mathbb{N}$ (en binaire bien entendu), et retournant en sortie le i-ième mot (dans l'ordre lexicographique par longueur[7]) ne satisfaisant pas P. (Cette machine existe car P est calculable.)

Soit x un mot tel que $P(x) = \texttt{false}$. Soit alors $i_x \in \mathbb{N}$ l'index de x dans l'odre lexicographique par longueur. Alors $(\mathcal{M}, i_x)$ est une description de x de longueur $\langle \mathcal{M}, i_x\rangle = |i_x| + c$ où $c = |\langle\langle \mathcal{M}\rangle\rangle| + 2$.

Soit $b > 0$. On choisit un $n \in \mathbb{N}$ tel que moins de $\frac{2^n}{2^{b+c}}$ mots de longueur n ou moins ne satisfassent pas P. (Cela est possible car P est vraie presque partout.) Alors, on a $i_x \leq \frac{2^n}{2^{b-c}}$.

Ainsi, on a $|i_x| \leq n - b - c$ et donc $|\langle M, i_x\rangle| \leq n - b$. D'où

$$K(x) \leq n - b$$

Un mot b-incompressible ne satisfaisant pas P a donc une longueur inférieure ou égale à n. Il y en a donc un nombre fini.

♣

[7]Les mots sont classés par longueur et, pour chaque longueur ,dans l'ordre lexicographique. Cet ordre permet que tout mot de longueur finie ait un index.

4.4 Ouverture sur les mots infinis...

Pour traiter du hasard raisonnablement, il faut passer des mots finis (représentation de $\mathbb{N}$) aux mots infinis (représentation de $\mathbb{R}$).

Bien que nous ne développons pas ici le sujet, on peut y définir une complexité similaire à celle de KOLMOGOROV (qui revient à la complexité de KOLMOGOROV sur les préfixes) et prolonger notre modélisation du hasard.

On *trouve* alors des mots incompressibles intéressants tels la constante Ω de CHAITIN définie comme suit : les machines de TURING étant dénombrables, on les dénombre $(\mathcal{M}_i)_{i\in\mathbb{N}}$; on note ensuite b_i le bit valant 1 si $\mathcal{M}_i$ s'arrête sur l'entrée vide, 0 sinon ; Ω est alors le nombre $0, b_1b_2b_3\cdots = \sum_{i\in\mathbb{N}} b_i \cdot 10^{-i}$.

Cette constante est un nombre bien défini, mais non calculable.

Références

[Car08] O. Carton. *Langages formels, calculabilité et complexité*. Vuibert, 2008.

[She00] A. Shen. Algorithmic Information Theory and Kolmogorov Complexity. Course's notes about Kolmogorov complexity, 2000.

[Sip96] M. Sipser. *Introduction to the Theory of Computation*. International Thomson Publishing, 1996.