LOGIQUE & CALCUL

Le défi des faibles complexités

Comme les très petites durées ou longueurs, les faibles complexités sont délicates à estimer. Paradoxalement, leur évaluation exige des calculs colossaux.

Jean-Paul DELAHAYE

Prenons un objet défini par un grand nombre de données, par exemple une photo numérique ou le texte d'un livre. Le bon sens nous souffle que plus grande est la quantité d'information nécessaire pour décrire l'objet, plus il est « complexe ». Le corollaire est que plus votre objet est complexe, plus sa transmission prendra du temps.

Si l'image à laquelle nous nous intéressons est toute noire (par exemple un carré d'un million de pixels noirs), la brève description « l'image a pour dimension 1 000 × 1 000 et tous ses pixels sont noirs » suffit pour la connaître avec une parfaite précision. Dans le cas d'une photographie de visage, même les meilleures méthodes de description exigeront des milliers de bits d'information. Le visage est plus complexe et la valeur de cette complexité déterminera le coût de la transmission.

Les théoriciens ont précisé cette façon naturelle et intuitive d'envisager la complexité d'une suite de 0 et de 1. C'est le concept de complexité introduit vers 1965 en même temps par Andreï Kolmogorov et Gregory Chaitin. La complexité de Kolmogorov d'un objet numérique s est la taille $K(s)$ du plus court programme informatique, une description optimale de s, qui, exécuté, reconstitue exactement s. La taille d'un fichier compressé de l'objet s est une mesure de sa complexité. En effet, l'algorithme de décompression, partant de ce fichier, reconstitue toutes les données de l'objet s (on n'envisage ici que la compression sans perte) ; nous considérons donc ce fichier compressé comme un programme dont la taille est approximativement celle du programme le plus court donnant s.

Étudiée depuis la décennie 1960, la complexité de Kolmogorov a été appliquée en physique, en biologie et même en économie. Elle permet de concevoir clairement ce qu'est le contenu en informations d'un objet.

Échec de la complexité de Kolmogorov

Avec la complexité de Kolmogorov, nous savons caractériser mathématiquement une suite aléatoire, ce dont était incapable la théorie classique des probabilités : une suite aléatoire s de n bits d'information (une suite de n fois 0 ou 1) est une suite dont la complexité de Kolmogorov $K(s)$ est environ n. Une suite infinie aléatoire est une suite dont aucun début (la suite des n premiers bits) n'est sensiblement compressible. Au contraire, la suite infinie 01010101... n'est pas aléatoire, car on peut définir de façon concise ses $2n$ premiers bits par « n fois 01 ».

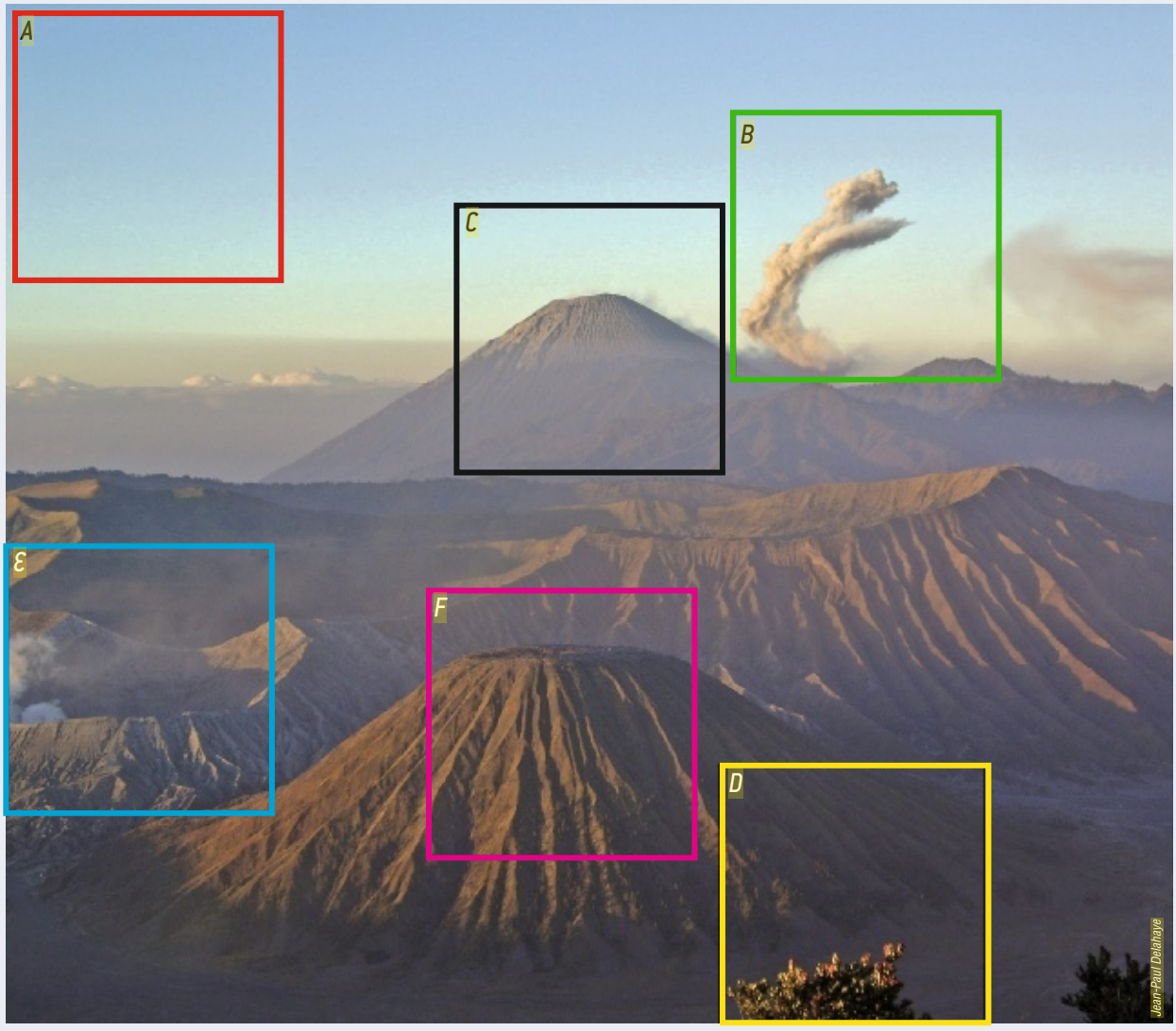
Le calcul des valeurs approchées de la complexité de Kolmogorov, grâce aux algorithmes de compression de données sans perte, rend le concept utilisable : la figure 1 propose des exemples de calcul de la complexité de Kolmogorov pour des images numériques. Certaines méthodes de classification de textes, de musiques et de séquences génétiques ont été déduites d'algorithmes de compression de données.

Cette théorie a cependant un défaut : elle ne permet de mesurer la complexité d'un objet s que s'il est assez grand (supérieur à quelques centaines de bits). La raison de cette limitation est que la définition de $K(s)$ dépend du langage de programmation utilisé pour obtenir la « longueur du plus court programme » : selon le langage choisi, la complexité de Kolmogorov subit des variations qui ont peu d'importance lorsqu'il s'agit de longues séquences de 0 et de 1, mais qui enlèvent tout sens aux résultats concernant les petites séquences.

Pourtant, il semble censé d'envisager la complexité de Kolmogorov des séquences courtes. En effet, il paraît clair que, par exemple, la séquence de sept bits 1001101 est plus complexe que la séquence 0000000, et que la séquence 0001000 est d'une complexité intermédiaire à celles des deux précédentes. Nous avons donc une idée intuitive d'un classement par complexité croissante des séquences courtes. Comment faire pour que la théorie donne un sens à ce que notre intuition perçoit, et que cette théorie s'applique à la fois aux séquences courtes et longues ?

Le problème à résoudre est un problème de thermomètre : parmi tous les instruments de mesure qui conduisent à des évaluations de la complexité de Kolmogorov, mais qui diffèrent par des constantes additives, n'y en aurait-il pas un meilleur, ou devant être préféré, au moins sur un plan

1. Comprimer pour mesurer



La complexité de Kolmogorov d'un fichier informatique *s* est la taille du plus court programme, mesurée en nombre de bits, capable de produire le fichier *s*.

Un algorithme de compression de données transforme un fichier informatique *s* en un fichier « compressé » *s'*. Certains algorithmes sont dits « sans perte », car lorsqu'on décompresse le fichier *s'*, celui-ci redonne exactement le fichier initial *s*. Ces algorithmes sans perte nous intéressent, car le fichier compressé *s'* peut être assimilé à un programme minimal. La taille du fichier compressé *s'* est donc une valeur approchée de la complexité de Kolmogorov du fichier initial *s*.

Voici une photo du volcan *Bromo* situé à l'Est de l'île de Java en Indonésie. Plusieurs parties de l'image de même taille géométrique (mesurée en nombre de pixels) ont été sélectionnées et confiées à un algorithme de compression de données sans perte, l'algorithme PNG.

La taille des fichiers compressés varie en fonction de la complexité de la partie de photo sélectionnée. Le ciel presque uniforme *(image A)* a une complexité de 324 kilo-octets (1 kilo-octet = 8 192 bits). Le panache de projections émis par le volcan *(image B)* est un peu plus complexe (400 kilo-octets).

Les autres parties sont encore plus complexes : l'image *C* a une complexité de 416 kilo-octets ; *D*, de 420 kilo-octets ; *E*, de 512 kilo-octets ; et *F*, de 544 kilo-octets.

2. Complexités

Certaines méthodes séduisantes utilisées pour le calcul de la complexité des courtes séquences sont insuffisantes. En effet, elles ne coïncident pas avec la complexité de Kolmogorov quand la longueur des séquences augmente, ce qui est nécessaire pour que la mesure soit satisfaisante.

L'entropie de Shannon est définie par l'expression $-\sum p_i \log p_i$ (où p_i est une fréquence). Avec une suite binaire, cela donne $[-p_1 \log p_1 - p_2 \log p_2]$, où p_1 et p_2 sont respectivement les densités de 0 et de 1 dans la séquence. Avec cette mesure, la suite 01010101010101010101 est la plus complexe possible pour une suite longueur 20 (elle a autant de 0 que de 1). Or la suite 10010111010100001011, qui a la même entropie de Shannon et la même longueur, est plus complexe. L'entropie de Shannon ne prend en compte que les fréquences de 0 et de 1 ; ce n'est pas suffisant pour savoir si une séquence est simple ou complexe. C'est une mesure statistique qui ne prend même pas en compte les répétitions.

La complexité par facteurs d'une suite s est déterminée par le nombre de facteurs différents qu'on trouve dans s. Un facteur est une suite de symboles contigus. Ainsi, la séquence 00000 possède cinq facteurs différents : 0, 00, 000, 0000, 00000. Elle a comme complexité 5. La séquence 01011 possède les 11 facteurs différents 0, 1, 01, 10, 11, 010, 101, 011, 0101, 1011, 01011 et elle est donc plus complexe.

Cette mesure est mauvaise, car elle attribue à certaines suites régulières une grande complexité. Ainsi, la suite de Champernowne obtenue en écrivant les entiers en binaire les uns derrière les autres (0 1 10 11 100 101 110 111 1000 etc.) met en défaut la mesure par facteurs. Par exemple, la suite 01101110010111011111000 (les entiers jusqu'à 8) n'est pas complexe, car on peut la définir brièvement et pourtant, elle contient un très grand nombre de facteurs. La mesure de complexité par facteurs ne peut pas être adoptée : une mesure générale doit être équivalente à la complexité de Kolmogorov lorsque les longueurs tendent vers l'infini, or ce n'est pas du tout le cas ici.

pratique ? L'instabilité des thermomètres les plus simples (ceux fondés sur la longueur minimale des programmes) les rend inutilisables pour le classement des faibles complexités.

La solution est apparue il y a environ cinq ans. Un thermomètre nouveau, mis au point au Laboratoire d'informatique fondamentale de Lille selon les idées et les programmes de Hector Zenil, donne des réponses satisfaisantes pour les courtes et les longues séquences. Il définit une complexité conforme à l'idée générale de Kolmogorov pour les grandes complexités et classe, comme on l'attend, 0000000, 0001000, 1001101, ou toutes les séquences pour lesquelles notre intuition n'hésite pas.

Le nouveau thermomètre que nous détaillerons possède un défaut : il est extrêmement coûteux en calcul. En pratique, il ne donne de résultats que pour les séquences très courtes et, de ce point de vue, les méthodes par compression restent incontournables. Comme en astronomie où, selon le type d'objets et leur éloignement, on utilise un procédé ou un autre pour évaluer les distances, dans les mesures de complexité, on doit adopter des méthodes hybrides avec recollement de plusieurs procédés.

Un critère fondé sur la probabilité de production

L'idée sous-jacente aux thermomètres pour les basses complexités est fondée sur une propriété remarquable de la complexité de Kolmogorov : plus un objet est simple, plus il est produit fréquemment quand on fait fonctionner un ordinateur en lui donnant des programmes choisis au hasard.

Un langage de programmation universel étant donné (*Java, C, Pascal, Basic* sont de tels langages), nous imaginerons que nous tirons au hasard des programmes écrits dans ce langage (en fait on peut écrire les programmes en binaire et tirer un programme au hasard revient à tirer une suite de 0 et de 1 jusqu'à avoir un programme complet). Nous tomberons souvent sur des programmes grammaticalement incorrects (donc ne fonctionnant pas) ou sur des programmes qui tourneront sans produire aucun résultat. Ces programmes ne nous intéressent pas : seuls ceux qui produisent en sortie une suite finie de 0 et de 1 et s'arrêtent sont pris en compte. Les expériences montrent qu'en procédant ainsi, on obtient plus souvent 0000000 que 1001101 : la probabilité d'obtenir en sortie une suite de grande complexité est plus faible que la probabilité d'obtenir une suite de faible complexité. Cette probabilité se nomme mesure de Solomonoff-Levin, en l'honneur de Ray Solomonoff (1926-2009) et Leonid Levin (professeur à l'Université de Boston), qui ont identifié ce phénomène.

Un théorème, clef des thermomètres pour les faibles complexités, fait le lien avec la complexité de Kolmogorov. Ce théorème affirme que $K(s)$ est environ égal à $-\log_2 SL(s)$, où $\log_2$ est le logarithme en base 2 (on a $\log_2(x) = \log x / \log 2$). En termes mathématiques, $K(s)$ étant une complexité de Kolmogorov définie par un langage de programmation et $SL(s)$ la mesure de Solomonoff-Levin associée au même langage, il existe une constante c, indépendante de la séquence s, telle que :

$$\left| -\log_2 SL(s) - K(s) \right| < c.$$

À l'infini, $-\log_2 SL(s)$ et $K(s)$ ne diffèrent au plus que d'une constante fixée une fois pour toutes. La quantité $-\log_2 SL(s)$ est donc aussi intéressante qu'une complexité de Kolmogorov mesurée par les plus courts programmes et coïncide presque avec elle pour les longues séquences.

Le premier avantage à considérer le nombre $-\log_2 SL(s)$ plutôt que $K(s)$ provient de ce que c'est un nombre bien plus stable. En changeant de langage de programmation, ou seulement en modifiant légèrement la définition d'un langage de programmation choisi, nous ferons beaucoup varier $K(s)$ pour les petites séquences, ce qui empêche, nous l'avons vu, de donner un sens à ce qu'on obtient. En revanche, le nombre $-\log_2 SL(s)$ ne varie que très peu. La raison est que ce nombre résulte d'un lissage opéré sur un très grand nombre de données dû au fonctionnement d'une multitude de programmes.

Cependant, l'argument le plus puissant qui fait s'intéresser à la mesure de Solo-