# What is Kolmogorov Complexity?

A. Shen[1,2]

[1] Институт проблем передачи информации РАН, Москва,
127994, Большой Каретный, 19
[2] LIF CNRS, Marseille;
E-mail: alexander.shen@lif.univ-mrs.fr,
shen@mccme.ru

Roughly speaking, Kolmogorov complexity is “compressed size”. Programs like zip, gzip, etc., compress a file (text, program, or some other data) into a presumably shorter one. The original file can then be restored by a “decompressing” program.

This explanation contains several inaccuracies—both technical and more essential. Technically, instead of files (sequences of bytes) we will consider binary strings (finite sequences of bits, that is, of zeros and ones).

Here are the more essential points:

- We consider only decompressing programs; we do not worry at all about compression. A *decompressor* is any algorithm (program) that receives a binary string as an input and returns a binary string as an output. If a decompressor $D$ on input $x$ terminates and returns string $y$, we write $D(x) = y$ and say that $x$ is a *description* of $y$ with respect to $D$. Decompressors are also called *description modes.*

- A description mode is not required to be total. For some $x$, the computation $D(x)$ may never terminate and therefore produces no result. There are no constraints on the computation time of $D$.

In other words, a description mode is a partial computable (=partial recursive) function from $\Xi$ to $\Xi$, where $\Xi$ stands for the set of all binary strings.

Assume that a description mode (a decompressor) $D$ is fixed. For a string $x$ consider all its descriptions, that is, all $y$ such that $D(y)$ is defined and equals $x$. The length of the shortest string $y$ among them is called the *Kolmogorov complexity* of $x$ with respect to $D$:

$$K_D(x) = \min\{\, l(y) \mid D(y) = x \}.$$

Here $l(y)$ denotes the length of the string $y$. The subscript $D$ indicates that the definition depends on the choice of the description mode $D$. The minimum of the empty set is $+\infty$, thus $K_D(x)$ is infinite for all the strings $x$ outside the range of the function $D$.

At first glance this definition seems to be meaningless, as for different $D$ we obtain quite different notions, including ridiculous ones. For instance, if $D$ is nowhere defined, then $K_D$ is infinite everywhere.

A more reasonable example: consider a decompressor $D$ that just copies its input to output, that is, $D(x) = x$ for all $x$. In this case every string is its own description and $K_D(x) = l(x)$.

Of course, for any given string $x$ we can find a description mode $D$ that is tailored to $x$ and with respect to which $x$ has small complexity. Indeed, let $D(\Lambda) = x$. This implies $K_D(x) = 0$.

It may seem that the dependence of complexity on the choice of the decompressor makes impossible any general theory of complexity. However, it is not the case.

A description mode is better when descriptions are shorter. We say that a description mode (decompressor) $D_1$ is *not worse* than a description mode $D_2$ if $K_{D_1}(x) \leqslant K_{D_2}(x) + c$ for some constant $c$ and for all strings $x$.

Let us comment on the role of the constant $c$ in this definition. We consider a change in the complexity bounded by a constant as “negligible”. One could say that such a tolerance makes the complexity notion practically useless, as the constant $c$ can be very large. However, nobody managed to get any reasonable theory that overcomes this difficulty and defines complexity with better precision.

The starting point for the algorithmic information theory was the following Kolmogorov – Solomonov universality theorem:

> *There is a description mode $D$ that is not worse than any other one: for every description mode $D'$ there is a constant $c$ such that $K_D(x) \leqslant K_{D'}(x) + c$ for every string $x$.*

A description mode $D$ having this property is called *optimal.*

An optimal description mode can be constructed as follows: $D(py) = p(y)$ where $p$ is a program (in some "self-delimiting" programming language, where one can find the end of the program while reading it from left to right) and $y$ is any binary string.

If $p$ is a program for some decomressor $P$, and $y$ is a shortest description of the string $x$ with respect to $P$ then $py$ is a description of $x$ with respect to $D$ (though not necessarily the shortest one) and $K_D(x) \leqslant K_P(x) + l(p)$.

Fix some optimal description mode $D$ and call $K_D(x)$ the *Kolmogorov complexity* of the string $x$. In the notation $K_D(x)$ we drop the subscript $D$ and write just $K(x)$.

Could we then consider the Kolmogorov complexity of a particular string $x$ without having in mind a specific optimal description mode used in the definition of $K(x)$? No, since by adjusting the optimal description mode we can make the complexity of $x$ arbitrarily small or arbitrarily large.

One may wonder then whether Kolmogorov complexity has any sense at all. Comparing two optimal decompressors based on different programming languages, we see that the difference in complexities is bounded by a constant that is the length of the program that is written in one of these two languages and interprets the other one. If both languages are "natural", we can expect this constant to be not that huge, just several thousands or even several hundreds. Therefore if we speak about strings whose complexity is, say, about $10^5$ (i.e., a text of a novel), or $10^6$ (DNA string) then the choice of the programming language is not that important.

**Some properties of Kolmogorov complexity**:

- *There is a constant $c$ such that $K(x) \leqslant l(x) + c$ for all strings $x$.*

- *For every algorithm $A$ there exists a constant $c$ such that*

  $$K(A(x)) \leqslant K(x) + c$$

  *for all $x$ such that $A(x)$ is defined.*

- *There is a constant $c$ such that $K(xy) \leqslant K(x) + 2\log K(x) + K(y) + c$ for all $x$ and $y$.*

- *Let $n$ be an integer. Then there are less than $2^n$ strings $x$ such that $K(x) < n$.*

Kolmogorov complexity of a string somehow measures the "amount of information" in this string. Before the algorithmic information theory, Shannon entropy was used as a measure of information. However, it could hardly be used for individual objects.

Assume that we want to use Shannon entropy to measure the amount of information contained in some English text. To do this we have to find an "ensemble" of texts and a probability distribution on this ensemble such that the text is "typical" with respect to this distribution. This makes sense for a short telegram, but for a long text (say, a novel) such an ensemble is hard to imagine.

The same difficulty arises when we try to define the amount of information in the genome of some species. If we consider as the ensemble the set of the genomes of all existing species (or even all species ever existed), then the cardinality of this set is rather small (it does not exceed $2^{1000}$ for sure). And if we consider all its elements as equiprobable (which other distribution can we choose?) then we obtain a ridiculously small value (less than 1000 bits).

So we see that in these contexts Kolmogorov complexity looks like a more adequate tool than Shannon entropy.

**Randomness.** Kolmogorov complexity is a natural way to formalize the intuitive notion of "randomness": random string is a string that is hard to compress. In other words, the difference between the length of a string $x$ and $K(x)$ could be considered as "randomness deficiency" of a string $x$.

To get a sharp boundary line between random and nonrandom objects we have to consider infinite sequences of zeros and ones instead of strings. A reasonable definition of randomness was given by P. Martin-Löf. Later L. Levin and C. Schnorr found a characterization of randomness in terms of complexity.

**Noncomputability of Kolmogorov complexity function.** It would be nice to be able to compute the Kolmogorov complexity of a given string by some algorithm. However, it is easy to see that this is not possible. The proof is a reformulation of the so-called "Berry's paradox". This paradox considers *the minimal natural number that cannot be defined by at most fourteen English words*; this phrase has fourteen words and defines that number.

Following this idea, imagine that Kolmogorov complexity is computable. Then we can test all the strings and find the *first binary string whose Kolmogorov complexity is greater than $N$*. By definition, its complexity is greater than $N$, but this string has a short description that includes the binary notation of $N$ (and the total number of bits requires is much less than $N$ for large $N$).

**Occam's razor and Kolmogorov complexity**. What do we mean when we say that a theory is a good explanation of some experimental data? Assume that we are given some experimental data and there are several theories to explain the data. For example, the data might be the observed positions of planets in the sky. We can explain them as Ptolemy did, with epicycles and deferents, introducing extra corrections when needed. On the other hand, we can use the laws of the modern mechanics. Why do we think that the modern theory is better? A possible answer: the modern theory can compute the positions of planets with the same (or even better) accuracy given less parameters. In other words, Kepler's achievement is a shorter description of the experimental data.

Roughly speaking, experimenters obtain binary strings and theorists find short descriptions for those strings (thus proving upper bounds for their complexities); the shorter the description is, the better is the theorist.

This approach is sometimes called "Occam's razor" and is attributed to the philosopher William of Ockham who said that entities should not be multiplied beyond necessity.

Taking this approach to a (rather absurd) extreme, one can announce the contest: participants have to provide the shortest possible program that prints the human genome string. (The more regularities we find in the genome, the shorter this program would be.)

**Information distance and classification**. The amount of mutual information in two strings $x$ and $y$ can be measured roughly as $I(x:y) = K(x)+K(y)-K(xy)$ (common information in $x$ and $y$ need not to be repeated in $xy$). We cannot compute this quantity, but we can—with no justification at all—replace $K$ by the compressed size (using some fixed compression program, such as `bzip`) and then use this value for classification of binary strings (say, DNA sequences or text files). Some initial experiments confirm the practical value of this approach.

**Further reading**. The first paper by Kolmogorov: Колмогоров А. Н., Три подхода к определению понятия к количество информацик, *Проблемы передачи информации*, 1965, т. 1, вып. 1, с. 3–11 (English translation: Kolmogorov A. N., Three approaches to the quantitative definition of information. *Problems Inform. Transmission*, 1(1):1–7, 1965.)

The most comprehensive text on the subject: Li M., Vitányi P., *An Introduction to Kolmogorov Complexity and Its Applications*, Second Edition, Springer, 1997. (638 pp.)

A short freely available lecture notes of an introductory course: A. Shen, *Algorithmic Information Theory and Kolmogorov Complexity*, published as Technical Report 2000-034, Uppsala universitet (available online). See also Russian book in preparation written by V. Uspensky, N. Vereshchagin and A. Shen (current version see `ftp.mccme.ru/user/shen/kolmbook.ps.gz`). Experiments of classification using compression are described in R. Cilibrasi, P. Vitanyi, *Clustering by compression*, http://arxiv.org/abs/cs.CV/0312044.

# What is Kolmogorov Complexity?

A. Shen[1,2]

[1] Институт проблем передаци информации РАН, Москва,
127994, Большой Каретный, 19
[2] LIF CNRS, Marseille;
E-mail: alexander.shen@lif.univ-mrs.fr,
shen@mccme.ru

Roughly speaking, Kolmogorov complexity is “compressed size”. Programs like zip, gzip, etc., compress a file (text, program, or some other data) into a presumably shorter one. The original file can then be restored by a “decompressing” program.

This explanation contains several inaccuracies—both technical and more essential. Technically, instead of files (sequences of bytes) we will consider binary strings (finite sequences of bits, that is, of zeros and ones).

Here are the more essential points:

- We consider only decompressing programs; we do not worry at all about compression. A *decompressor* is any algorithm (program) that receives a binary string as an input and returns a binary string as an output. If a decompressor $D$ on input $x$ terminates and returns string $y$, we write $D(x) = y$ and say that $x$ is a *description* of $y$ with respect to $D$. Decompressors are also called *description modes.*

- A description mode is not required to be total. For some $x$, the computation $D(x)$ may never terminate and therefore produces no result. There are no constraints on the computation time of $D$.

In other words, a description mode is a partial computable (=partial recursive) function from $\Xi$ to $\Xi$, where $\Xi$ stands for the set of all binary strings.

Assume that a description mode (a decompressor) $D$ is fixed. For a string $x$ consider all its descriptions, that is, all $y$ such that $D(y)$ is defined and equals $x$. The length of the shortest string $y$ among them is called the *Kolmogorov complexity* of $x$ with respect to $D$:

$$K_D(x) = \min\{\, l(y) \mid D(y) = x \}.$$

Here $l(y)$ denotes the length of the string $y$. The subscript $D$ indicates that the definition depends on the choice of the description mode $D$. The minimum of the empty set is $+\infty$, thus $K_D(x)$ is infinite for all the strings $x$ outside the range of the function $D$.

At first glance this definition seems to be meaningless, as for different $D$ we obtain quite different notions, including ridiculous ones. For instance, if $D$ is nowhere defined, then $K_D$ is infinite everywhere.

A more reasonable example: consider a decompressor $D$ that just copies its input to output, that is, $D(x) = x$ for all $x$. In this case every string is its own description and $K_D(x) = l(x)$.

Of course, for any given string $x$ we can find a description mode $D$ that is tailored to $x$ and with respect to which $x$ has small complexity. Indeed, let $D(\Lambda) = x$. This implies $K_D(x) = 0$.

It may seem that the dependence of complexity on the choice of the decompressor makes impossible any general theory of complexity. However, it is not the case.

A description mode is better when descriptions are shorter. We say that a description mode (decompressor) $D_1$ is *not worse* than a description mode $D_2$ if $K_{D_1}(x) \leqslant K_{D_2}(x) + c$ for some constant $c$ and for all strings $x$.

Let us comment on the role of the constant $c$ in this definition. We consider a change in the complexity bounded by a constant as “negligible”. One could say that such a tolerance makes the complexity notion practically useless, as the constant $c$ can be very large. However, nobody managed to get any reasonable theory that overcomes this difficulty and defines complexity with better precision.

The starting point for the algorithmic information theory was the following Kolmogorov – Solomonov universality theorem:

> *There is a description mode $D$ that is not worse than any other one: for every description mode $D'$ there is a constant $c$ such that $K_D(x) \leqslant K_{D'}(x) + c$ for every string $x$.*

A description mode $D$ having this property is called *optimal*.

An optimal description mode can be constructed as follows: $D(py) = p(y)$ where $p$ is a program (in some "self-delimiting" programming language, where one can find the end of the program while reading it from left to right) and $y$ is any binary string.

If $p$ is a program for some decomresssor $P$, and $y$ is a shortest description of the string $x$ with respect to $P$ then $py$ is a description of $x$ with respect to $D$ (though not necessarily the shortest one) and $K_D(x) \leqslant K_P(x) + l(p)$.

Fix some optimal description mode $D$ and call $K_D(x)$ the *Kolmogorov complexity* of the string $x$. In the notation $K_D(x)$ we drop the subscript $D$ and write just $K(x)$.

Could we then consider the Kolmogorov complexity of a particular string $x$ without having in mind a specific optimal description mode used in the definition of $K(x)$? No, since by adjusting the optimal description mode we can make the complexity of $x$ arbitrarily small or arbitrarily large.

One may wonder then whether Kolmogorov complexity has any sense at all. Comparing two optimal decompressors based on different programming languages, we see that the difference in complexities is bounded by a constant that is the length of the program that is written in one of these two languages and interprets the other one. If both languages are "natural", we can expect this constant to be not that huge, just several thousands or even several hundreds. Therefore if we speak about strings whose complexity is, say, about $10^5$ (i.e., a text of a novel), or $10^6$ (DNA string) then the choice of the programming language is not that important.

**Some properties of Kolmogorov complexity**:

- *There is a constant $c$ such that $K(x) \leqslant l(x) + c$ for all strings $x$.*
- *For every algorithm $A$ there exists a constant $c$ such that*

  $$K(A(x)) \leqslant K(x) + c$$

  *for all $x$ such that $A(x)$ is defined.*
- *There is a constant $c$ such that $K(xy) \leqslant K(x) + 2\log K(x) + K(y) + c$ for all $x$ and $y$.*
- *Let $n$ be an integer. Then there are less than $2^n$ strings $x$ such that $K(x) < n$.*

Kolmogorov complexity of a string somehow measures the "amount of information" in this string. Before the algorithmic information theory, Shannon entropy was used as a measure of information. However, it could hardly be used for individual objects.

Assume that we want to use Shannon entropy to measure the amount of information contained in some English text. To do this we have to find an "ensemble" of texts and a probability distribution on this ensemble such that the text is "typical" with respect to this distribution. This makes sense for a short telegram, but for a long text (say, a novel) such an ensemble is hard to imagine.

The same difficulty arises when we try to define the amount of information in the genome of some species. If we consider as the ensemble the set of the genomes of all existing species (or even all species ever existed), then the cardinality of this set is rather small (it does not exceed $2^{1000}$ for sure). And if we consider all its elements as equiprobable (which other distribution can we choose?) then we obtain a ridiculously small value (less than 1000 bits).

So we see that in these contexts Kolmogorov complexity looks like a more adequate tool than Shannon entropy.

**Randomness**. Kolmogorov complexity is a natural way to formalize the intuitive notion of "randomness": random string is a string that is hard to compress. In other words, the difference between the length of a string $x$ and $K(x)$ could be considered as "randomness deficiency" of a string $x$.

To get a sharp boundary line between random and nonrandom objects we have to consider infinite sequences of zeros and ones instead of strings. A reasonable definition of randomness was given by P. Martin-Löf. Later L. Levin and C. Schnorr found a characterization of randomness in terms of complexity.

**Noncomputability of Kolmogorov complexity function**. It would be nice to be able to compute the Kolmogorov complexity of a given string by some algorithm. However, it is easy to see that this is not possible. The proof is a reformulation of the so-called "Berry's paradox". This paradox considers *the minimal natural number that cannot be defined by at most fourteen English words*; this phrase has fourteen words and defines that number.

Following this idea, imagine that Kolmogorov complexity is computable. Then we can test all the strings and find the *first binary string whose Kolmogorov complexity is greater than $N$*. By definition, its complexity is greater than $N$, but this string has a short description that includes the binary notation of $N$ (and the total number of bits requires is much less than $N$ for large $N$).

**Occam's razor and Kolmogorov complexity**. What do we mean when we say that a theory is a good explanation of some experimental data? Assume that we are given some experimental data and there are several theories to explain the data. For example, the data might be the observed positions of planets in the sky. We can explain them as Ptolemy did, with epicycles and deferents, introducing extra corrections when needed. On the other hand, we can use the laws of the modern mechanics. Why do we think that the modern theory is better? A possible answer: the modern theory can compute the positions of planets with the same (or even better) accuracy given less parameters. In other words, Kepler's achievement is a shorter description of the experimental data.

Roughly speaking, experimenters obtain binary strings and theorists find short descriptions for those strings (thus proving upper bounds for their complexities); the shorter the description is, the better is the theorist.

This approach is sometimes called "Occam's razor" and is attributed to the philosopher William of Ockham who said that entities should not be multiplied beyond necessity.

Taking this approach to a (rather absurd) extreme, one can announce the contest: participants have to provide the shortest possible program that prints the human genome string. (The more regularities we find in the genome, the shorter this program would be.)

**Information distance and classification.** The amount of mutual information in two strings $x$ and $y$ can be measured roughly as $I(x:y) = K(x)+K(y)-K(xy)$ (common information in $x$ and $y$ need not to be repeated in $xy$). We cannot compute this quantity, but we can—with no justification at all—replace $K$ by the compressed size (using some fixed compression program, such as `bzip`) and then use this value for classification of binary strings (say, DNA sequences or text files). Some initial experiments confirm the practical value of this approach.

**Further reading.** The first paper by Kolmogorov: Колмогоров А. Н., Три подхода к определению понятия к количество информациик, *Проблемы передачи информации*, 1965, т. 1, вып. 1, с. 3–11 (English translation: Kolmogorov A. N., Three approaches to the quantitative definition of information. *Problems Inform. Transmission*, 1(1):1–7, 1965.)

The most comprehensive text on the subject: Li M., Vitányi P., *An Introduction to Kolmogorov Complexity and Its Applications*, Second Edition, Springer, 1997. (638 pp.)

A short freely available lecture notes of an introductory course: A. Shen, *Algorithmic Information Theory and Kolmogorov Complexity*, published as Technical Report 2000-034, Uppsala universitet (available online). See also Russian book in preparation written by V. Uspensky, N. Vereshchagin and A. Shen (current version see `ftp.mccme.ru/user/shen/kolmbook.ps.gz`). Experiments of classification using compression are described in R. Cilibrasi, P. Vitanyi, *Clustering by compression*, http://arxiv.org/abs/cs.CV/0312044.