Kolmogorov Complexity

Computational Complexity Course Report

By

Henry Xiao

Queen's University

School of Computing

Kingston, Ontario, Canada

March 2004

Introduction

In computer science, the concepts of algorithm and information are fundamental. So the measurement of information or algorithms is crucial in sense of describing. In 1965 *Andrey Nikolaevich Kolmogorov* [O'Connor, and Robertson, 1999], a Russian mathematician, established the algorithmic theory of randomness via a measure of complexity, now referred to *Kolmogorov complexity*. According to Kolmogorov, the complexity of an object is the length of the shortest computer program that can reproduce the object. All algorithms can be expressed in programming language based on *Turing machine* models equally succinctly, up to a fixed additive constant term. The remarkable usefulness and inherent rightness of the theory of Kolmogorov complexity or so called *Descriptive complexity,* stems from this independence of the description method.

The idea of Kolmogorov complexity first appeard in the 1960's in papers by Kolmogorov, Solomonoff and Chaitin. As specified by Schöning and Randall, an algorithm can exhibit very different complexity behavior in the worst case and in the average case. The Kolmogorov complexity is defined a probability distribution under which worst-case and average-case running time (for all algorithm simultaneously) are the same (up to constant factors). Quick sort algorithm has been widely taken as an example to show the applicability of Kolmogorov complexity since the algorithm takes $O(n \log n)$ time in average but $\Omega(n^2)$ time at worst case. Later, the Kolmogorov complexity was connected with *Information Theory* and proved to be closely related to *Claude Shannon*'s entropy rate of an information source. The theory base of Kolmogorov complexity has also be extended to data compression and communication for the sake of true information measure.

Kolmogorov Complexity Theory

We will briefly take a look at Kolmogorov Complexity definition and some main related results at this section. For details, please refer to [Cover and Thomas, 1991] and [Schöning and Pruim 1998].

Definition: The Kolmogorov Complexity $K_{\mathcal{U}}(x)$ of a string x with respect to a universal computer $\mathcal{U}$ is defined as:

$$K_{\mathcal{U}}(x) = \min_{p:\ \mathcal{U}(p) = x} l(p),$$

the minimum length over all programs that print x and halt. Thus $K_{\mathcal{U}}(x)$ is the shortest description length of x over all descriptions interpreted by computer $\mathcal{U}$. *(Note Turing machine is regarded as universal computer in computer science.)*

The concept of Kolmogorov Complexity asks for the minimal unambiguous description of a sequence. It can be used to prove complexity lower bounds. [Schöning and Pruim 1998] has two examples of using this technique. And indeed, as mentioned, the proofs obtained in this way are much more "elegant", or at least shorter, than the original proofs. In a few cases, the lower bounds were first achieved by means of Kolmogorov Complexity. Another quite important definition is the *conditional Kolmogorov complexity* which is based on the knowledge of the length of x, denoted as $l(x)$.

$$K_{\mathcal{U}}(x \mid l(x)) = \min_{p:\ \mathcal{U}(p,\, l(x)) = x} l(p),$$

This is the shortest description length if the computer $\mathcal{U}$ has the length of x made available to it. Quite a few different results have been shown for conditional Kolmogorov complexity too.

We look at some basic and interesting properties of Kolmogorov complexity and then consider some examples. The proofs of the theorems are eliminated because of the purpose of this report. However, for your interests, please refer to [Cover and Thomas, 1991] for complete proofs.

Both the lower and upper bounds of Kolmogorov complexity of a given sequence have been derived early. There are some choices of both bounds from different aspects. The followings are all established theorems for bounds.

Universality of Kolmogorov complexity: If $\mathcal{U}$ is a universal computer, then for any other computer $\mathcal{A}$: $\mathbf{K_u(x) \le K_A(x) + c_A}$ for all string $x \in \{0, 1\}^*$, where the constant $\mathbf{c_A}$ does not depend on x.

Conditional complexity is less than the length of sequence: $\mathbf{K_u(x|\, \mathit{l}(x)) \leqslant \mathit{l}(x) + c.}$

Upper bound on Kolmogorov complexity: $\mathbf{K_u(x) \leqslant K_u(x|\, \mathit{l}(x)) + 2\log(\mathit{l}(x)) + c.}$

Lower bound on Kolmogorov complexity: The number of strings x with complexity

$\mathbf{K_u(x) \le k}$ satisfies: $|\{ x \in \{0, 1\}^* : \mathbf{K_u(x) \le k} \}| < 2^k$.

The Kolmogorov complexity of a binary string x is bounded by:

$$\mathbf{K}(x_1x_2x_3\ldots x_n|n) \leqslant n\mathbf{H}_0(1/n \textstyle\sum_i x_i) + 2\log n + \mathbf{c},$$

where $H_0(p) = -p \log p - (1-p) \log (1-p)$ is the binary entropy function.

The above five theorems are basically considered to be very important facts of Kolmogorov complexity. Many other interesting results have been derived applying these theorems. We consider some examples of Kolmogorov complexity here to show the usability of the theory with its properties. First, we will look at some intuitive ideas directly coming from the definition. *A sequence of n zeros* has a constant Kolmogorov complexity, i.e. K(*000…0*|*n*) = *c*, since if we assume *n* is known, the a short program can directly print out *n* zeros. The same case can be applied to π, where the first n bits of π can be calculated using a simple series expression. A somewhat surprising result for *fractal* is that regarding its complex calculations, it is still essentially very simple in terms of Kolmogorov complexity which is nearly zero. An integer on the other hand, has higher complexity even it looks very straight forward. It is obviously true that the complexity of describing an integer will be constant if we know the length of the integer, i.e. K(*n*|*l(n)*) = *c*. However, in general, the computer does not know the length of binary representation of the integer. So we must inform the computer in some way when the description ends. We can bound the description using the upper bound we got so far: K(*n*) $\leqslant$ 2log *n* + c. We can also prove there are an infinite number of integers *n* such that K(*n*) > log *n*. This is probably less intuitive than we thought. From above examples, it is not hard to see the true measurement of information or algorithm would be rather hard without the development of Kolmogorov complexity. With the support of the Kolmogorov complexity theory, one can describe information more accurate in computer science.

Kolmogorov complexity also applied to algorithmically random, incompressible sequences, the halting problem, etc. We can not go into details of those examples; instead, we briefly look at those problems here just to get some taste. Many literatures have been

published on Kolmogorov complexity related questions. One can refer to [Li, and Vitanyi, 1999] and [Yaniv, 2003] for future reading. The algorithmically random and incompressible sequences are defined based on the Kolmogorov complexity properties that some sequences hold. We say a sequence $x_1, x_2, x_3 \ldots x_n$ algorithmically random if $K(x_1x_2x_3\ldots x_n| n) \geqslant n$. And we can say a string x incompressible if $lim\ K(x_1x_2x_3\ldots x_n| n)/n = 1$. The definitions seem very intuitive with respect of Kolmogorov complexity. In fact, if every element of the sequence is completely generated in random, we can not predict any later elements from current. Indeed, we will need to describe each element separately. However, if the Kolmogorov complexity verse n approaching 1 for a string in probability, then we can actually interpret this as the proportions of 0's and 1's in this string are almost equal, which is ½. By this meaning, it is true that we can not compress the string since any bit will be a critical contribution to the whole, which also specifies the randomness of the string. The next significant application is on the halting problem. Using Kolmogorov complexity, we can actually demonstrate that the problem can not be solved by an algorithm because of the non-computability of Kolmogorov complexity. It is rather a surprising fact of the non-computability. However, practically speaking, one may never be able to tell the shortest program since there are infinite many programs for a given sequence. We can only estimate the complexity by running more and more programs, as we know the bound will converge to the Kolmogorov complexity. Many other results can also be found on published literatures related with probability theory and information theory. At the following section, we take a look at one of the most important results related with the central idea of information theory – *Entropy*.

Kolmogorov Complexity and Entropy

As mentioned at introduction, the Kolmogorov complexity and entropy of a sequence of random variables are highly related. In general, the expected value of the Kolmogorov complexity of a random sequence is to its entropy.

From information theory founded by Shannon, the true measure of information on random variables is entropy. This relationship actually proves the correctness of the Kolmogorov complexity as a measurement of information and algorithms. Kolmogorov complexity states the shortest description (program) for a random variable. Then complexity of the sequence constructed by random variables will approach to the expect value of the set of Kolmogorov complexities for each variable, i.e. $E[1/n\mathrm{K}(X^n|n)] \rightarrow \mathrm{H}(X)$, supported by the law of large numbers in probability. Respectively, the information measure of the sequence is indeed entropy. Actually, one can always show the program lengths satisfy prefix condition, since if the computer halts on any program, it does not look any further for input. The relationship can be shown further with *Kraft inequality*.

The relationship provides us to two ways of complexity measures, which either takes after Kolmogorov complexity, involving finding some computer or abstract automaton which will produce the pattern of interest, or take after information theory and produce something like the entropy, which, while in principle computable, can be very hard to calculate reliably for experimental systems. This is actually very powerful principle in physics (see [Li, and Vitanyi, 1999] for details). A more interesting idea about "Occam's Razor" as a general principle governing scientific research can be derived from Kolmogorov theory as we will see at the next section.

Occam's Razor

I decide to include this topic mainly because its importance plus the idea really amazed me. At 14th century, William of Occam (Ockham in some literatures), a logician, said "Nunquam ponenda est pluralitas sine necessitate", i.e., explanations should not be multiplied beyond necessity, which forms the basis of *methodological reductionism*. Here, our argument will be a special case of it.

Recent papers have suggested a connection between Occam's Razor and Kolmogorov complexity. Many literatures take Laplace's sun rising problem as an example to explain the connection. Laplace considered the probability that the sun will rise again tomorrow, given that it has risen every day in recorded history. He solved it before Kolmogorov complexity was introduced. However, the problem can be reconsidered through Kolmogorov complexity. If we use 1 to represent the sun rise, then the probability that the next symbol is a 1 given n 1's in the sequence so far is: $\sum_y p(1^n 1y) \approx p(1^\infty) = c > 0$. And the probability that the next symbol is 0, which means that the sun will not rise: $\sum_y p(1^n 0y) \approx p(1^n 0) \approx 2^{-\log n}$, since any $1^n 0\ldots$ yields a description of n with length at least K(n), i.e. about log n + O(log log n). Hence the conditional probability of observing a 0 next is: $p(0|1^n) = p(1^n 0) / (p(1^n 0) + p(1^\infty)) \approx 1/(cn+1)$. The result is very similar to $1/(n+1)$ derived by Laplace. The Kolmogorv complexity solution to this question is actually following the Occam's Razor by weighting possible explanations by their complexity.

It often happens that the best explanation is much more complicated than the simplest possible explanation because it requires fewer assumptions. *Albert Einstein*

wrote in 1933 “Theories should be as simple as possible, but no simpler.” In our case, Kolmogorov complexity does provide us an alternative approach to explain things in many science fields.

Conclusion

Kolmogorov complexity is a profound theory for information and algorithm measure. The theory is somehow different from others that we have studied in computational complexity so far. I feel the theory tries to observe the complexity from a new approach. It is worth reading something about Kolmogorov [O'Connor, and Robertson, 1999] to understand his original idea, which he developed from mathematical perspective. It is in high level of abstraction, but closely related with many things in practice. Vladimir V’yugin presents some applications of Kolmogorov complexity in his review [V’yugin, 1994] mathematically. In computing, I found many ongoing topics related with Kolmogorov complexity are on information process ([Levin, 1999] and [Wallace, and Dowe, 1999a]). Generally, the application of Kolmogorv complexity is based on framework of the *Minimum Description Length (MDL)* principle and *Minimum Message Length (MML)* principle, which are out the scope of this report, but refer to [Wallace, and Dowe, 1999b] for introductions.

I strongly believe the Kolmogorov complexity should have more appearances in future research topics. Along with information theories, we need to deal with information as well as problems coming with it. Like the Shannon’s entropy theory established today’s communication system, the closely related Kolmogorov complexity shall also be of great potential to be applied to future researches.

Reference

Cover, M. Thomas, and Thomas, A. Joy. *Elements of information theory*. John Wiley & Sons, Inc. 1991.

Gammerman, Alexander, and Vovk, Vladimir. *Kolmogorov complexity: sources, theory and applications*. The Computer Journal, vol. 42, no. 4, 1999.

Levin, A. Leonid. *Robust Measures of Information*. The Computer Journal, vol. 42, no. 4, 1999.

Li, Ming, and Vitanyi, Paul. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer-Verlag, New York, 1999.

O'Connor, J J, and Robertson, E F. *"Andrey Nikolaevich Kolmogorov"*. School of Mathematics and Statistics, University of St. Andrews, Scotland, 1999.

Rissanen, Jorma. *Discussion of paper `Minimum Message Length and Kolmogorov Complexity' by C. S. Wallace and D. L. Dowe*. The Computer Journal, vol. 42, no. 4, 1999.

Schöning, Uwe, and Pruim, Randall. *Gems of theoretical computer science*. Springer-Verlag Berlin Heidelberg 1998.

Szabo, Nick. "*Introduction to Algorithmic Information Theory*". http://szabo.best.vwh.net/kolmogorov.html. 1996.

V'yugin, V. V. *Algorithm entropy (complexity) of finite objects and its applications to defining randomness and amount of information*. Selecta Mathematica Sovietica, 13, 357-389, 1994.

Wallace, S. C., and Dowe, L. D. *Minimum Message Length and Kolmogorov Complexity*. The Computer Journal, vol. 42, no. 4, 1999a.

Wallace, S. C., and Dowe, L. D. *Refinements of MDL and MML Coding*. The Computer Journal, vol. 42, no. 4, 1999b.

Yaniv, El Ran. "*Topics in Kolmogorov Complexity, Seminar 236804*". Computer Science Dept. Israel Institute of Technology. 2003.

Dear Dr. Salomaa,

I would like to get some suggestion or feedback from you after you read my report. It would be appreciated if you could send your comments to me by e-mail.

E-mail: xiao@cs.queensu.ca

Sincerely,

Henry Xiao