

Representing Information with Computational Resource Bounds

Daby Sow and Alexandros Eleftheriadis
Department of Electrical Engineering
Columbia University
New York, NY, 10027, USA
`{daby,eleft}@ee.columbia.edu`

Abstract

A general framework for data compression, in which computational resource bounds are introduced at both the encoding and decoding end, is presented. We move away from Shannon's traditional communication system by introducing some structure at the decoder and model it by a Turing machine with finite computational resources. Information is measured using the resource bounded Kolmogorov complexity. In this setting, we investigate the design of efficient lossy encoders.

1 Introduction

The problems of *quantifying representing and transmitting* information have been addressed in 1948 by C. E. Shannon [10] in a pure communication setting. The result of this work is Information Theory (IT), a mathematical basis formalizing the communication problem between a sender and a receiver. In this framework, the meaning of the message is irrelevant and completely ignored. The main question is to find an efficient representation of the output of a stochastic information source. This representation must be short enough to fit in the channel capacity, robust enough to survive the corruption of noise and accurate enough to provide a good approximation of the original message at the receiving end. This theory is based on probability theory. The notion of information commonly called entropy, is an *ensemble* one measuring the number of possible choices to make in order to select a message. Ignoring the effect of noise, optimality is then synonymous to compression which is achieved by assigning shorter codes to messages with higher probability. Targeted applications involved vertical designs such as telegraphy, telephony, facsimile, videoconferencing, or even digital television.

Today's picture of the communication world is however much broader. The ever increasing power of modern computers has transformed them into very capable platforms for audiovisual content creation and manipulation. Users today can easily capture compressed audio, images, or video, using a wide array of consumer electronic products. (e.g., digital image and video cameras as well as PC boards that produce JPEG and MPEG-1 content directly). It is quickly realized that the traditional objective of efficiency may conflict with other applications requirements (ease of editing, processing, indexing and searching, etc.).

There is an increasing need to develop systems able to *understand* information content in individual messages and it is not natural to add this new component in Shannon's framework where semantics are ignored. In 1965, in an attempt to measure the amount of randomness in individual objects, Kolmogorov introduced another information measure based on length of descriptions¹ [8]. In this case, entropy is a lack of compressibility measured individually by the length of the shortest computer program able to generate the object to represent. As shown in figure 1, this approach is a deterministic dual to Shannon's theory. Clearly, the understanding question is better addressed in this individualistic framework. It is important to note that short descriptions are desired here not only for transmission through a channel with limited capacity. Objects with long descriptions are patternless and intuitively random. Furthermore, following the Occam's razor principle, short descriptions are desired because they are simple and more likely to be good explanations for the object to describe. These descriptions are computer programs formalized using the theory of recursive functions [7]. The structure of the decoding device is fixed. It is a computer modeled by a Turing machine (TM). Interestingly, and in contrast with traditional IT, computational resource bounds can be introduced naturally at the decoding end by limiting the computational resources of the decoding TM. The result is a formalization of the practical problem of coding *finite* objects with only a finite amount of computational resources.

In this paper, we investigate the lossy representation of information in Kolmogorov's setting with computational resource bounds at both the decoding and encoding end. We start in section 2, with a brief exposition of definitions and notations used throughout this work. In section 3, we show the duality between this deterministic approach and Shannon's stochastic approach. This section will highlight the main similarities and differences between these two theories and will give the motivation for the use of Kolmogorov's

¹Solomonoff and Chaitin also introduced independently the same notion, respectively in 1964 and 1966 with different motivations. Solomonoff formalized the notion of Occam's razor for inductive inference. Chaitin measured the complexity of a string by the smallest number of states a Turing machine needs to output the string.

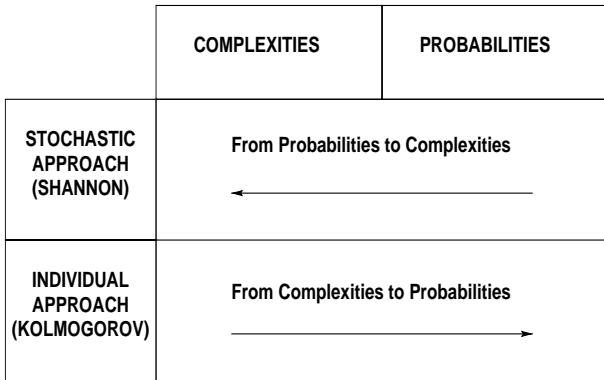


Figure 1: Duality between Shannon and Kolmogorov information measures.

approach to the coding of finite objects. In section 4, we discuss the design of universal codec systems for finite objects with a finite amount of computational resources. A general method based on evolutionary programming techniques is presented.

2 Notations and Definitions

Without any loss of generality we focus here on binary sequences. Let $B_0 = \{0, 1\}$, $B_0^n = B_0 \times \dots \times B_0$ n times and $B = \bigcup_{n=1}^{\infty} B_0^n$. B^∞ denotes the set of all infinite binary sequences (the continuum). The empty string is denoted by Λ . Let n and m be two elements of \mathcal{N} and $x \in B$, we define l as being a function from B to \mathcal{N} mapping each element x of B to its length. x_m denotes the m^{th} symbol x . x_n^m will denote $(x_n, x_{n+1}, \dots, x_m)$ if $m > n$.

2.1 Kolmogorov Complexity

The concept of Kolmogorov complexity is formalized using the theory of recursive functions. A function f mapping elements of B to B is recursive if there is a Turing machine able to take any element x of B on one of its input tapes and map it to $f(x) \in B$ by having $f(x)$ on its output tape².

Definition 1 [8] Let F be an arbitrary partial recursive function with a prefix domain³. Then the prefix conditional complexity of the sequence $x \in B$ given the knowledge of another sequence y with respect to F is:

$$K_F(x | y) = \begin{cases} \min\{l(p) : F(p, y) = x\}, \\ \infty \text{ if } \forall p \in B \ F(p, y) \neq x \end{cases}$$

$$K_F(x) = K_F(x | \Lambda)$$

Theorem 1 There exists a partial recursive function F_0 (called optimal) such that for any other partial recursive function G ,

$$K_{F_0}(x | y) \leq K_G(x | y) + O(1) \quad (1)$$

²See [7] for a complete discussion on this concept.

³A prefix domain is a set of sequences where no sequence is the proper prefix of another.

Proof: See [8]. □

The optimal function is also called the universal function. The intuition behind this theorem is the existence of a universal computer able to simulate the actions of any other computer. As a result, we will drop the subscript referring to the partial recursive function and use $K(x) = K_{F_0}(x)$ as a notation for the complexity of sequence x .

Theorem 2 K is not partial recursive.

Proof: See [5], [9]. □

Theorem 2 is known as the Noncomputability Theorem. It is a negative results since it proves that any attempt to compress maximally *any* sequence cannot be performed on a TM. It is a manifestation of the Halting problem but fortunately, the next theorem states that it is possible to approximate K .

Theorem 3 There is a total recursive function $\Phi(t, x)$, monotonic decreasing in t , such that

$$\lim_{t \rightarrow \infty} \Phi(t, x) = K(x)$$

Proof: This theorem is due to Kolmogorov (according to [9]). Its proof can be found in [9] and [5]. Since the proof hints at how computational resource bounds at the decoding end remove the Halting problem, we reproduce it here. For each value of t , it is possible to construct a prefix Turing machine that runs the reference Turing machine fixed in theorem 1 for no more than t steps on each program with length at most $l(x) + c$. We define $\Psi(t, \cdot)$ as the partial recursive function computed by this Turing machine. If for some input programs p , the computation halts with output x , then we define $\Phi(t, x) = \min\{l(p) : \Psi(t, p) = x\}$. Else, $\Phi(t, x) = l(x) + c$. Clearly, $\Phi(t, x)$ is recursive, total⁴, and nonincreasing with t . Also, the limit exists since for each x , there is a t such that the reference Turing machine halts and outputs x after t steps starting with input p with⁵ $l(p) = K(x)$. □

Theorem 3 will play a major role in this paper. Although we can approximate $K(x)$ from above, it does not mean that we can decide whether $\Phi(t, x) = K(x)$ or not. In more intuitive words, the approximation gets closer and closer to K but it is impossible to know how close it gets. The proof of this theorem motivates the following definition.

Definition 2 [5] Let $\Phi^{t,s}$ be a partial recursive function computed by a Turing machine such that for any $x \in B$, the computation of $\Phi^{t,s}(x)$ requires less than t

⁴Every finite x has a value assigned to it by Φ .

⁵Since objects are finite, there is a value t^* corresponding to the number of steps the optimal description for x (with length equal to $K(x)$) takes when it is placed in input on the reference Turing machine.

steps (time) and uses less than s tape cells (memory). The resource-bounded Kolmogorov complexity $K_{\Phi}^{t,s}$ of x , conditional to Φ and y ($y \in B$), is defined by

$$K_{\Phi}^{t,s}(x | y) = \min \{l(p) : \Phi^{t,s}(p, y) = x\} \quad (2)$$

Theorem 1 can be extended to the resource-bounded Kolmogorov complexity. As a consequence, we can drop the subscript Φ and denote this complexity by $K^{t,s}$.

2.2 Complexity Distortion Function

The Kolmogorov complexity has been extended to the lossy case in [12] and [11]. In this section, we go one step further and put some resource bounds on the decoding Turing machine. To do so, we first generalize the optimal quantization procedure proposed in [11].

Definition 3 Let D be a distortion measure according to a single letter fidelity criterion⁶ $d(\cdot, \cdot)$. On a given object x_1^n , with D as a constrain, we introduce distortion in order to minimize the resource-bounded complexity of the resulting object y_1^n . If we have more than one object y_1^n with the same optimal complexity, we select the closest to x_1^n . If many objects y_1^n are equidistant to x_1^n , we arbitrarily select one of them and map it to x_1^n . This way, we define a function from the set of source objects x_1^n to the set of distorted objects y . We denote this function $\mathcal{Q}_D^{t,s}$ and call it the optimal quantization procedure⁷.

$$\mathcal{Q}_D^{t,s}(x_1^n) = \arg \min_{y_1^n \in B^n : d(x_1^n, y_1^n) \leq D} K^{t,s}(y_1^n)$$

Definition 4 The resource-bounded complexity distortion function is a mapping from B to \mathcal{R}^+ associating elements $x_1^n \in B$ to:

$$C_D^{t,s}(x_1^n) = \frac{K(\mathcal{Q}_D^{t,s}(x_1^n))}{n} \quad (3)$$

The complexity distortion function $C_D(\cdot)$ is defined by:

$$C_D(x_1^n) = \lim_{s, t \rightarrow \infty} C_D^{t,s}(x_1^n) \quad (4)$$

From theorem 3, $C_D(\cdot)$ is well defined.

3 Duality between $C_D(\cdot)$ and $R(D)$

To claim that Shannon and Kolmogorov's approaches are dual to each other, it is imperative find some relationships between these information measures. This closes the circle of media representation techniques shown in figure 2. These equivalences are well known in the lossless case [5],[9]. The lossy case is discussed in [11],[12].

⁶See [3]

⁷For simplicity, we assume that the range of $\mathcal{Q}_D^{t,s}$ is also B . The results still holds for the more general ranges for $\mathcal{Q}_D^{t,s}$, with the definition of the minimum amount of achievable distortion. See [3].

Theorem 4 [11] For any stationary ergodic source with a recursive probability measure μ , let $R(D)$ be the rate distortion function associated with a single letter distortion measure, then,

$$\lim_{n \rightarrow \infty} C_D(x_1^n) = R(D), \quad x_1^n \in B^n \quad (5)$$

μ -almost surely.

Proof Outline: To prove theorem 4, first we use Markov types and the code construction proposed in [6] to establish an upper bound on $C_D(\cdot)$. The Markov k -type of x_1^n , denoted by $\hat{p}_k(x_1^n)$, is a measure of the empirical distribution of x_1^n , assuming a k -order Markov model. The type class $T_k(x_1^n)$ of x_1^n is the set of all sequences of length n with Markov k -type equal to $\hat{p}_k(x_1^n)$. We represent x_1^n with a two part code. The first part of the code represents the Markov k -type of x_1^n . To do so, an index of the Markov k -type of x_1^n in the class of all possible Markov k -types is transmitted. We call this part the model of the representation. The second part of the code is called the data part. To compute it, we first partition $T_k(x_1^n)$ into distortion classes. Each of these classes is composed of sequences distant from each other by less than D . The partition is done in order to minimize the total number of distortion classes needed to cover $T_k(x_1^n)$. The second part of the code is then an index to the distortion class that contains⁸ x_1^n . It can be shown that this code is prefix free and can be understood by a prefix TM. Its coding rate converges almost surely to $R(D)$ [6]. In the second step of the proof, we use the incompressibility of almost all infinite sequences (using integral randomness tests⁹) to show that most sequences have their Kolmogorov complexity close to their Shannon entropy. More formally, in a lossless context, let the set of typical sequences, S_{typ} , be defined as:

$$S_{typ} = \left\{ x \in B^\infty : \liminf_{n \rightarrow \infty} \frac{-\log_2 \mu(x_1^n) - K(x_1^n)}{n} = 0 \right\}$$

If μ is recursive then,

$$\mu(S_{typ}) = 1 \quad (6)$$

Note that this result coupled with the Shannon-McMillan-Breiman theorem shows that for a stationary ergodic source, the ratio complexity length converges to the entropy rate. In the lossy case, we use definition 3 and equation 6 to argue that the mutual information between the source and the output of the optimal quantizer (defined by $\mathcal{Q}_D^{t,s}(\cdot)$) is almost surely equal to the rate distortion function¹⁰. It remains to show that this rate is almost surely equal to the complexity distortion function and this can be done using the definition of the mutual information (as a Radon-Nikodym derivative [3]) and equation 6 again.

⁸We assume that the output of the optimal quantizer belongs to B .

⁹See [5] chap 4.

¹⁰This statement is a direct consequence of the optimality of $\mathcal{Q}_D^{t,s}$ and the almost surely equivalence between complexities and logarithmic of probabilities.

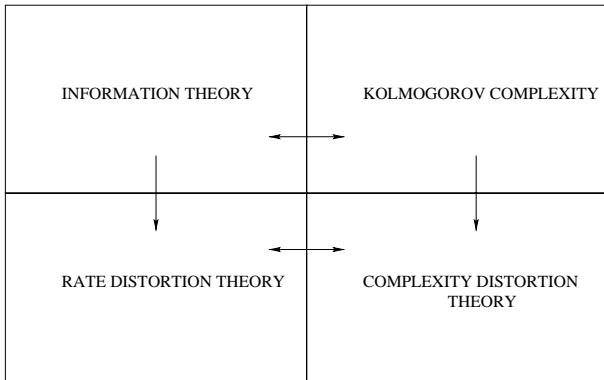


Figure 2: The circle of media representation Theories.

□

There are two interesting point to make from this outline. First, restricting the decoding function to be recursive does not reduce the performances if the source has a recursive probability measure. In fact the Church-Turing thesis guarantees that any coding algorithm belongs to the set of recursive functions, from traditional entropy coding techniques to model-based coding methods. The result is a unification of all coding algorithms under a single framework. Second, the equivalence was made possible by using limits showing that Shannon's information measure assumes the availability of an infinite amount of computational resources at the decoding end. Furthermore, due to its ensemble nature, this information measure requires infinite observations which may not be available in practice. Its dual part, the Kolmogorov complexity, is not suitable for the prediction of compression rates when computational resources are unbounded as shown in theorem 2 (where Shannon's approach works well) but in contrast with Shannon's measure, it predicts recursive compression rates for the coding of finite individual objects with a finite amount of computational resources.

4 Codec Design

As we saw in section 3, convergence between complexity distortion function and rate distortion function is due to the existence of types or relative frequencies. Classical coding techniques use this property and are only asymptotically optimal. The existence of the limit is guaranteed by the assumption that the source is stationary and ergodic. In a sense, the class of ergodic sources contains the most general dependent source for which the law of large numbers holds. In this section we propose a coding system that do not rely on statistical properties of the source. Instead, a complexity approach is used. We will argue that this coding system yields performances arbitrary close to the resource bounded complexity distortion function.

4.1 The Decoder

The key component of the complexity distortion approach is the substitution of the decoder in Shannon's classical communication system by a Turing machine

completely defined by its language. Designing this language delimits the space of possible representations that we call the program search space. With a Turing complete language, optimal representations yielding compression rates close to the complexity distortion function belongs to the search space. But such a complete space could be too wide and inefficient for a particular application. Prior knowledge on the application, can be used to limit the dimensions of the search space and speed up the encoding procedure. To simplify the discussion, we assumed that our language is Turing complete. Also, without loss of generality, we represent programs using either symbolic expressions¹¹ or parse trees,

4.2 The Encoder

With the program space specified by the structure of the decoder, the encoder has to explore this space and find an efficient representation for the source object to be coded. An interesting way to perform this search is to use evolutionary programming techniques, like genetic programming. The idea is to perform a beam search which is a compromise between exhaustive and hill climbing techniques[2]. An evaluation metric, commonly called a fitness measure, is used to measure the efficiency of each point in the program space. This number is a measure of how well the corresponding program represents the source object. When lossy compression with resource bounds at the decoder is the main problem, the fitness has to be a function of the amount of distortion introduced by the representation and the length of the representation. The fitness is used to select out a certain number of the most promising solutions for further transformation. The genetic programming method starts by generating randomly an initial population of programs, also called a generation. These programs are then run for less than t steps and using less than s memory cells to determine their fitness. Using these fitness numbers for this population, and following the Darwin principle of survival of the fittest, a new generation of programs is obtained by performing genetic operations. The most common operations are the crossover, the mutation and the reproduction. In the crossover operation, two parent programs belonging to the initial generation are chosen. Subtrees of these programs are randomly chosen and swap to give birth to two offsprings in the new generation. Parents with high fitness have a higher probability to participate in crossover operations. The mutation operation simply changes randomly some nodes in the parse trees of individuals of the new generation. The reproduction copies good programs in the new generation. Details of these operations can be found in [2]. What is interesting here is that under general conditions (to be mentioned below), when this process is repeated, the probability to have an element with maximum fitness in the population converges to 1 [4]. To see this, note that the dynamic of this algorithm can be modeled

¹¹Any computer program can be seen as a composition of functions. In fact compilers use this fact and first translate the code into a parse tree before translating it to machine code instructions.

by a Markov chain. Populations have fixed size, each possible one corresponding to a state in the Markov chain. Since the object that has to be coded is finite in length, the number of possible states in this process is finite¹². The convergence of the genetic process depends on the structure of the transition matrix Q of this Markov chain. As shown in [4], optimality can be reached almost surely in polynomial time if the following two points are satisfied:

1. The second largest eigenvalue¹³ of Q , denoted λ_{max} , is suitably bounded away from 1 so that the Markov chain is rapidly mixing.
2. The stationary distribution π gives probability greater than ϵ , where $\frac{1}{\epsilon}$ is polynomial in the problem parameter, to the set of states that contains individuals of best fitness.

The first property requires that Q is irreducible with non negative entries which will always be the case if we have a strictly positive mutation probablity forcing ergodicity. The second property is more difficult to satisfy. It can be ensured by a good design of the decoder. Assuming that it also holds, the following algorithm can be used at the encoder:

1. From a start state, evolve through a polynomial number of generations;
2. From the final population vector, select the fittest individual.
3. Repeat step 1 and 2 a polynomial number of times.

The third step of the algorithm is used to boost the convergence probability. Almost surely discovery of an individual with optimal fitness is guaranteed. This procedure, although polynomial time, is computationally expensive in practice. A similar version of it has been implemented for the coding of sounds and images in [1] using binary machine code to reduce the computational burden at the encoder. The use of machine code does in fact speed up the fitness evaluations during the encoding, which is the bottle neck of the algorithm. More work will done in this direction.

5 Concluding Remarks

In this paper, we have substitute the decoder in Shannon's classical communication system by a Turing machine. It allows us to place computational resource bounds at the decoding end. Information is measured using the resource bounded Kolmogorov complexity, which is asymptotically equivalent to Shannon's entropy. It shows the duality between Kolomogorov complexity theory and Shannon's theory of communication, which assumes the availability of an infinite amount of computational resources

¹²Formally, there is a constant c such that for all $x_1^n \in B_0^n$, $K(x_1^n) \leq n + c$. Therefore, the cardinal of the program space is bounded.

¹³The largest eigenvalue of Q is 1 if the chain is irreducible. Its associated left eigenvector is then π , the stationary distribution.

at the decoder. Finally, we have discussed the design of optimal codec systems for finite objects with limited decoding power using evolutionary programming methods.

References

- [1] P. Nordin and W. Banzhaf, "Programmatic Compression of Images and Sound," *Proceedings of the First Annual Conference on Genetic Programming*. J. R. Koza, D. E. Goldberg, D. B. Fogel and R. L. Riolo ed., The MIT Press, pp. 345-350, July 28-31 1996.
- [2] W. Banzhaf, P. Nordin, R. E. Keller and F. D. Francone, "Genetic Programming, An Introduction," *Morgan Kaufmann Publishers, Inc.* 1998.
- [3] T. Berger, "Rate Distortion Theory, a Mathematical Basis for Data Compression," *Prentice-Hall, Inc.* 1971.
- [4] P. Vitanyi, "A Discipline of Evolutionary Programming," *Theoretical Computer Science*, to appear.
- [5] M. Li and P. Vitanyi, "An Introduction to Kolmogorov Complexity and Its Applications (2nd Edition)," *Springer-Verlag, Inc.* 1997.
- [6] P. C. Shields, "Universal almost sure Data Compression using Markov Types," *Probl. of Control and Inform. Theory*, Vol 19(4), pp 269-277, 1990.
- [7] M. Davis, "Computability and Unsolvability," *Dover Publications, Inc.* 1982.
- [8] A. N. Kolmogorov, "Three Approaches to the Quantitative Definition of Information," *Probl. Inform. Transmission*, 1(1) pp 1-7, 1965.
- [9] A. K. Zvonkin and L. A. Levin, "The Complexity of Finite Objects and the Development of the Concepts of Information and Randomness by Means of the Theory of Algorithms," *Russian Math. Surveys*, Vol 25(6) pp 83-124, 1970.
- [10] C. E. Shannon, "A Mathematical Theory of Communication," *Bell Systems Technical Journal*, Vol 27, 1948.
- [11] D. Sow and A. Eleftheriadis, "Complexity Distortion Theory," *Proceedings 1997 IEEE International Symposium on Information Theory, Ulm Germany*, pp. 188, June 29 - July 4 1997.
- [12] E. H. Yang and S. Y. Shen, "Distortion Program-Size Complexity with Respect to Fidelity Criterion and Rate-Distortion Function," *Transactions on Information Theory*, Vol. 39, No 1, pp. 288-292, January 1993.