Complexidade de Kolmogorov e suas aplicações

Fernanda Maria de Souza

Joinville, Santa Catarina

Abstract

Em 1963, Andrei Nikolaevich Kolmogorov desenvolveu uma teoria da informação e da aleatoriedade baseada na descoberta, em 1936, da Máquina de Turing Universal por Alan Turing. A teoria prova que a complexidade de qualquer string binária é o tamanho do menor programa (ou descrição algorítmica) que pode produzir essa string em uma Máquina de Turing Universal e parar. Denominada de Complexidade de Kolmogorov, a mesma define uma nova teoria da informação, chamada teoria algorítmica da informação. De forma a conceituar os tópicos propostos, neste artigo será abordado, de forma geral, a história, explicação acerca do tema, definições auxiliares sobre propriedades relacionadas, fatos que comprovam sua importância e, por fim, aplicações da Complexidade de Kolmogorov.

Keywords: Complexidade de Kolmogorov, Teoria Algorítmica da Informação, Máquina de Turing, Aleatoriedade, Computabilidade

1. Introdução

Segundo [1], um dos aspectos mais importantes na ciência moderna é a complexidade. Kolmogorov, Solomonoff e Chaitin, de forma independente, definiram o que hoje conhece-se como Complexidade de Kolmogorov [2].

O conceito de complexidade de Kolmogorov está definido sobre o conjunto de strings binárias (sequências de zeros e uns). A mesma, associa a cada string binária um valor numérico que é a sua complexidade. Para [3], a complexidade de Kolmogorov pode ser definida simplificadamente como o tamanho do menor programa (ou descrição algorítmica) que computa em uma Máquina de Turing uma determinada string binária. A complexidade de Kolmogorov define uma nova teoria da informação chamada teoria algorítmica da informação.

A noção para a teoria surgiu quando Kolmogorov buscava obter uma definição formal sobre sequências aleatórias, e, a partir disso, acreditou que sequências poderiam ser comprimidas algoritmicamente através de regularidades e descritas com menos informações [2]. O mesmo detectou que sequências que não podem ser comprimidas algoritmicamente são sequências aleatórias, ou seja, que não possuem regularidades em sua descrição.

O método proposto esse trabalho baseia-se em desenvolver uma pesquisa bibliográfica relacionada a complexidade de Kolmogorov. São apresentadas as principais definições, teoremas e provas sobre a complexidade de Kolmogorov na Seção 3. Algumas de suas propriedades, na Seção 4. Aplicações são apresentadas na Seção 5. Finalmente, são apresentadas na Seção 6 e 7, respectivamente, a importância da complexidade de Kolmogorov e as considerações finais da pesquisa.

2. Histórico

Na década de 1960, no estágio inicial da ciência da computação com a teoria geral de máquinas de Turing sendo aprofundada, cientistas precisavam mensurar a computação e a informação quantitativamente [4]. Foi então que em 1965, a partir de seu famoso artigo [5], Andrei Nikolaevich Kolmogorov propôs a ideia de medir a quantidade de informação de objetos finitos utilizando abordagens algorítmicas. O objetivo original do trabalho de Kolmogorov era obter uma definição formal de sequências aleatórias [3]. Kolmogorov observou que sequências binárias poderiam ser comprimidas algoritmicamente.

Ideias similares foram exploradas antes por Ray Solomonoff, porém de maneira diferente, o mesmo definiu a noção de probabilidade a priori. Em 1965, Gregory Chaitin com apenas 18 anos propôs a mesma definição de complexidade algorítmica de Kolmogorov.

Logo, a complexidade de Kolmogorov teve sua origem quando Andrei Kolmogorov, Ray Solomonoff e Gregory Chaitin desenvolveram, de forma independente, uma teoria baseada no tamanho dos programas para Máquina de Turing.

O principal impulso da teoria da complexidade de Kolmogorov foi a sua universalidade. A mesma foi desenvolvida buscando métodos de aprendizado universais baseados em métodos de codificação universais. Essa abordagem foi originada por Solomonoff e conseguiu mais visibilidade com a comunidade matemática graças a Kolmogorov.

2.1. Raízes da Complexidade de Kolmogorov

- Pierre-Simon Laplace: Uma sequência é extraordinária (não aleatória) porque contém rara regularidade [6].

- Richard von Mises: noção de uma sequência aleatória S:

$$\lim_{n\to\infty}\{\#(1)\ em\ n-prefixo\ de\ S\}/n = p,\ 0 < p < 1$$

  A definição acima é válida para qualquer subsequência de S selecionada por uma função "admissível" [7].

- Abraham Wald: O Lema de Wald. É uma identidade importante que simplifica o cálculo do valor esperado da soma de um número aleatório de quantidades aleatórias [8].

- Alonzo Church: funções de seleção recursiva [9].

- Teoria da informação de Shannon-Weaver[10].

- Inferência Bayseana: consiste na avaliação de hipóteses pela máxima verossimilhança, uma decorrência imediata da fórmula de Bayes [11].

- Máquina de Turing.

3. Definição

A complexidade de Kolmogorov, proposta por Kolmogorov-Solomonoff-Chaitin como uma teoria algorítmica da informação e aleatoriedade, é uma teoria profunda e sofisticada que trata da quantidade de informação de objetos individuais, medida através do tamanho de sua descrição algorítmica.

A teoria algorítmica da informação, de acordo com Chaitin, é "o resultado de colocar a teoria da informação de Shannon e teoria da computabilidade de Turing em uma coqueteleira, agitando-a vigorosamente" [12].

A teoria do que seria futuramente a complexidade de Kolmogorov, iniciou-se quando Kolmogorov observou que algumas sequências binárias poderiam ser comprimidas algoritmicamente. O mesmo postulou que aquelas que não podem ser comprimidas algoritmicamente em uma descrição de tamanho muito menor que a sequência original, são sequências determinadas aleatórias. Assim, definiu-se as sequências simples como sendo aquelas que são regulares ou compressíveis, e as strings aleatórias ou complexas como sendo aquelas que possuem irregularidade (são incompressíveis) [13].

Logo, a complexidade de Kolmogorov C(s) de qualquer *string* binária $s \in \{0,1\}$ é o tamanho do menor programa de computador p que pode produzir essa *string* em uma Máquina de Turing Universal MTU e parar [14]. Em outras palavras, pode-se interpretar que MTU C(s) *bits* de informação são necessários para codificar s, com a MTU relacionada a complexidade de Kolmogorov como um dispositivo de descompressão de dados.

Formalmente, a definição matemática da complexidade de Kolmogorov é dada pela Equação 1, onde l(p) é o tamanho (em bits) do programa p. Considera-se que $\phi : \{0,1\}^* -> \{0,1\}^*$ seja uma função recursiva parcial

$$C_\phi(s) = min\ l(p),\ p : \phi(p) = s, \tag{1}$$

A teoria surge como um caminho para descrever a aleatoriedade de strings e objetos finitos, tentando responder uma questão fundamental: "O que é um objeto aleatório?". Sejam as seguintes *strings* binárias, conforme Tabela 1:

Strings	**Binário**
1	1111111111
2	1100001010

Table 1: Strings binárias.

Com base na intuição humana, a maioria das pessoas consideraria que apenas a *string* 2 é aleatória, pois não é possível extrair nenhum padrão dela. Pela perspectiva da probabilidade, todas as duas *strings* tem a mesma probabilidade de serem escolhidas quando se têm uma *string* de 8 dígitos. Exemplificando a definição de Kolmogorov, a primeira *string* pode ser computada pelo programa:

$$print\ \text{"1"} * 8$$

Enquanto a segunda *string* pode ser computada como:

$$print\ \text{"1100001010"}$$

Uma vez que a noção de aleatoriedade está conectada com padrões em *strings* e o modo que podemos descrevê-las, a primeira *string* possui uma descrição algorítmica curta, e o tamanho do programa para n bits é de $\mathcal{O}(log\ n)$.

A segunda *string* (aleatória) possui uma longa descrição conforme sua irregularidade, e o tamanho do programa para computar *strings* desse tipo para n *bits* seria $n + c$ (c *bits* para a rotina que imprime), ou $\mathcal{O}(n)$ (aproximadamente o tamanho da própria *string*). Essa situação demonstra a Equação 1, onde a complexidade de determinada *string* é o tamanho de sua menor descrição.

A complexidade de Kolmogorov depende invariavelmente da escolha da linguagem descritiva da MTU [14]. Felizmente, o Teorema a seguir, nomeado Teorema da Invariância, proposto por Solomonoff-Kolmogorov [15][13], traz alguma ordem ao caos:

Teorema 1. Existe uma função recursiva parcial U que para qualquer função recursiva parcial ϕ se tem uma constante $c > 0$ como:

$$C_U(s) \leq C_\phi(s) + c \tag{2}$$

para todas strings s.

Uma máquina U que satisfaça o Teorema 1 é, em certo sentido, mínima entre todas as máquinas, e é chamada de universal [15] e [13].

Prova: Seja $\phi_0, \phi_1, \phi_2, ...$ uma enumeração de todas as funções parciais e $\langle s, x \rangle : \{0,1\}^* \times \{0,1\}^* - > \{0,1\}^*$, e.g., $\langle s, x \rangle = 0^{|s|}1sx$. U é definido como: na entrada w, decodifique w para dentro de i e p de tal modo que w = $\langle i, p \rangle$ e rode ϕ_i na entrada p. Se $\phi_i(p)$ parar então imprima o que ϕ_i têm para imprimir. Com isso, é possível verificar de forma que uma U é recursiva parcialmente e que satisfaz o Teorema 1.

Com o Teorema da Invariância, considera-se que a complexidade de Kolmogorov de uma *string* s como sendo $C_U(s)$. Para facilitar o raciocínio das definições, entretanto, o símbolo usado será $C(s)$.

Logo, desse modo, uma *string* s é aleatória (c-incompressível) se:

$$C(s) \geq |s| \tag{3}$$

Teorema 2: Existe ao menos $2^n - 2^{n-c} + 1$ c-incompressíveis *strings* de tamanho n.

Prova: Apenas existem $\sum_{k=0...n-c-1} 2^k = 2^{n-c} - 1$ programas com tamanho menor que n-c. Consequentemente, apenas esse número de *strings* (de um total de 2^n *strings* de tamanho n) podem ter programas (descrições) mais curtas do que n-c.

Também, é possível identificar que a definição de uma *string* aleatória não pode ser vazia, visto que existe uma *string* aleatória de Kolmogorov de cada tamanho: têm-se $2^n - 1$ descrições de tamanho menor que n, mas há 2^n *strings* de tamanho n.

Considerando algumas *strings* e sua complexidade de Kolmogorov:

- 0^n possui complexidade de Kolmogorov de $\log n + \mathcal{O}(1)$ pois só precisamos especificar o inteiro n e um curto programa que reconstruirá 0^n de n.

- y $\in \{0,1\}^{\sqrt{n}}$ que é Kolmogorov aleatório (c-incompressível). Em seguida, $x = y0^{n-\sqrt{n}}$ que possui complexidade de Kolmogorov acerca de $\sqrt{n} + \mathcal{O}(1)$. Caso houvesse uma descrição muito mais curta do que $|y|$, poderíamos descrever y usando essa descrição: primeiro produza x e então imprima apenas os primeiros $\sqrt{|x|}$ *bits*. De outro modo, a descrição de y é uma boa descrição de x: produza y e então acrescente $|y|^2 - |y|$ zeros. Logo, existem *strings* de essencialmente todas as complexidades.

4. Propriedades

Além das definições básicas sobre a complexidade de Kolmogorov, serão exploradas algumas propriedades principais da mesma nesta Seção.

4.1. Complexidade de conjuntos de Kolmogorov

Proposição 1: Seja A um conjunto recursivo (recursivamente enumerável) e n um inteiro, com $A_n = A \cap \{0,1\}^n$. Para todas as *strings* x em A_n, as mesmas possuem:

$$C(x) \leq \log |A_n| + 2\log n + \mathcal{O}(1) \tag{4}$$

Prova: A prova é direta. Visto que A é recursiva, pode-se elaborar um programa que de acordo com i e n escreve a i-ésima *string* de A_n. Consequentemente, toda as *strings* em A_n podem ser descritas pelo i, n e o programa para enumerar A.

A Proposição 1 revela o fato de que o conjunto de strings que não são Kolmogorov aleatórias, são recursivamente enumeradas, ou seja, com uma *string* x pode-se executar todos os programas de tamanho menor que x paralelamente para descobrir se algum deles escreve x. Isso acontece aceitando x.

Portanto, o número de *strings* de tamanho n que são Kolmogorov aleatórias é Kolmogorov aleatório por si só, sendo cerca de $2^n/c$ para alguma constante c < 1. Isso leva a determinação de que o número de *strings* não-aleatórias é 2^n menos o número de *strings* aleatórias, i.e., pode-se facilmente calcular um do outro, e também prova que a maioria das *strings* possui alta complexidade, não possuindo muitas *strings* com baixa complexidade de Kolmogorov [16].

4.2. Incomputabilidade

A complexidade de Kolmogorov não é computável [2]. O fato de que a complexidade de Kolmogorov não pode ser computada decorre do fato de que não podemos computar a saída de cada programa. Mais fundamentalmente, nenhum algoritmo é possível para prever cada programa e descobrir se algum dia os mesmos vão parar, como foi mostrado por Alan Turing no problema da parada [17].

Dado qualquer programa de computador como entrada, não é possível produzir uma saída *true* se esse programa parar ou *false* se não. Mesmo tendo-se um programa curto que produz nossa *string* e que parece ser um bom candidato para ser o programa mais curto, há sempre uma série de programas mais curtos dos quais não sabemos se algum dia vão parar e com que saída [18].

Definindo e provando formalmente que a complexidade de Kolmogorov é incomputável:

Teorema 3: A complexidade de Kolmogorov da função $w- > C(w)$ é incomputável.

Prova: Supondo o contrário, que C é computável e que UTM é uma máquina de Turing que o computa. Constrói-se uma nova máquina de Turing UTM' que computa *strings* de alta complexibilidade, mas UTM' terá uma breve descrição, formando uma contradição.

Especificamente, UTM' itera sobre um conjunto de *strings* binárias na ordem lexicográfica. Para cada *string* w, é computado C(w), parando uma vez que encontra w tal que $C(w) \geq |w| = n$. Então, a seguinte inequação é formada:

$$n \leq C(w) \leq |\langle UTM', n\rangle| + c \tag{5}$$

O motivo da desigualdade é apenas o Teorema da Invariância: $\langle UTM', n\rangle$ é uma descrição de w na linguagem das máquinas de Turing. Em outras palavras, a máquina de Turing universal que simula UTM' em n produzirá w, de modo que a complexidade de Kolmogorov de w é limitada pelo tamanho dessa descrição (mais uma constante).

Logo, o tamanho de $\langle UTM', n\rangle$ é no máximo $\log(n) + |\langle UTM'\rangle| + c'$ para alguma constante c', e isso por sua vez é $\log(n) + c''$ para alguma constante c". Por conseguinte, a inequação é dada:

$$n \leq log(n) + c^n \tag{6}$$

Mas, como $\log(n) = \mathrm{o}(n)$, pode-se escolher n suficientemente grandes para chegar a uma contradição.

5. Aplicações

Conforme a Seção 4.2 prova, a complexidade de Kolmogorov é incomputável. Porém, é possível por meio de estimativas chegar próximo ao número exato da complexidade. Portanto, as aplicações demonstradas na Seção 5 são exemplificações do uso dessas estimativas.

A seguir, são aprofundadas as aplicações de complexidade de Kolmogorov para Criptografia e Garantia da Informação, Filtragem de Spam e Fadiga Mental. Além disso, também é dada uma prova alternativa com a complexidade de Kolmogorov para o teorema clássico da existência de infinitos números primos. Ao final, uma pequena comparação entre a Navalha de Occam e a complexidade de Kolmogorov é citada.

5.1. Criptografia e Garantia da Informação

Uma das aplicações da complexidade de Kolmogorov é usando a mesma para detectar comportamentos anormais em sistemas de informação. O artigo [19] apresenta a ideia explorando o fato relacionado a questões de segurança da informação, que são geralmente tratadas após a detecção da violação de segurança. Com sistemas sendo invadidos, cavalos de troia são colocados, as senhas são descobertas e firewalls são quebrados. Por isso, o autor propõe a ideia de lidar com uma brecha de segurança no momento que a mesma acontece.

A partir da Complexidade de Kolmogorov, pode-se definir a complexidade de determinada *string* y em relação a outra *string* x. O conceito é que a informação de y possa ser usado para definir um programa p que calcule x [2]. A complexidade de Kolmogorov é descrita a partir da equação abaixo:

$$K\phi(y|x) = \begin{cases} min_{\phi(p,x)=y}\ l(p) \\ \qquad\qquad\quad \infty \quad \text{, se não existir p tal que} \quad \phi(p,x) = y \end{cases} \tag{7}$$

onde l(p) representa o tamanho de programa p e ϕ é um computador universal em consideração.

Portanto, a entrada de uma *string* x pode reduzir a complexidade ou tamanho do programa necessário para produzir uma nova *string* y[19].

Com base na Equação 1 e 7, o autor explica que uma menor complexidade de C(X,Y) tornará mais fácil para um invasor entender o que o sistema está fazendo (uma vez que suas entradas e saídas não são muito complexas). A outra métrica é monitorar a $C(Y|Z)$, se o sistema adiciona complexidade a X para produzir Y, então o sistema será menos vulnerável; se o sistema subtrair a complexidade de X para produzir Y, então o sistema ficará mais vulnerável. A relação dessas métricas de complexidade está descrita na Figura 1.

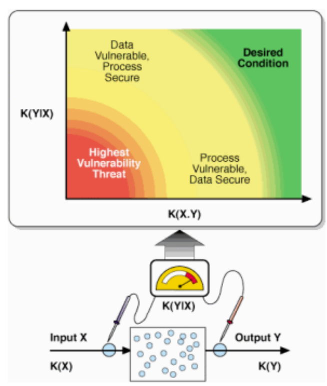

Figure 1: Vulnerabilidades de processo vs. dados.

Monitorando essas duas métricas e classificando a vulnerabilidade com base nas regiões mostradas, foi proposto o método de usar a complexidade de Kolmogorov para monitorar informações em um sistema, obtendo resultados satisfatórios sendo um bom candidato para pesquisas futuras nesta área.

5.2. Filtragem de Spam

Outra aplicação interessante da complexidade de Kolmogorov é a filtragem de *spams*. Conforme o artigo [20] revela, é possível por meio de um filtro adaptativo e estimativas da complexidade de Kolmogorov, fazer checagens se *emails* são *spams* ou comuns. Os autores fazem uso de um algoritmo de compressão para descobrir quanto a *string* pode ser compactada, usando esse dado como um limite superior na complexidade de Kolmogorov [2].

O estudo apontou que uma baixa complexidade de Kolmogorov indica uma maior probabilidade daquela mensagem ser um *spam*. Os dados estatísticos foram coletados mostrando a relação entre a complexidade de Kolmogorov e o *spam*, conforme Figura 2.

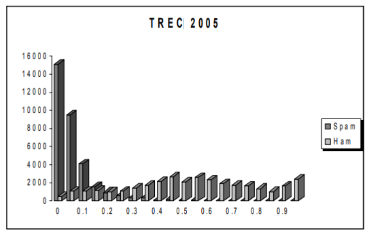

Figure 2: Histograma comparando a complexidade de Kolmogorov e a quantidade de *spams*.

Formalmente, o método usado no trabalho é definido:

$$K_C(x) = \frac{Tamanho(C(x))}{Tamanho(x)} + q \tag{8}$$

Sendo x a *string*, C o algoritmo de compressão, $K_c(x)$ a complexidade de Kolmogorov e q o tamanho em *bits* do programa que implementa C.

Portanto, segundo o artigo, concluiu-se que existe uma forte relação entre a complexidade de Kolmogorov e o tipo de *email* (comum ou *spam*).

5.3. Fadiga Mental

Visto que a complexidade de Kolmogorov opera em strings, a mesma pode realmente ser aplicada a qualquer coisa que possa ser descrita como uma *string*. Em [21], o autor aplica a complexidade de Kolmogorov aos sinais de EEG (Eletroencefalograma) obtidos a partir de pacientes, a fim de medir o nível de fadiga. O EEG é usado para detectar as atividades elétricas do sistema neural central (SNC).

Conforme descreve o artigo, os sinais foram medidos e transformados em sequências discretas, para então ser aplicado o método descrito em [22], em que a complexidade de Kolmogorov do objeto é estimada pelo número de etapas no processo de geração.

Formalmente, o algoritmo de complexidade de Kolmogorov C(n) foi dado a partir das Equações 9 e 10.

$$b(n) = \lim_{n \to \infty} c(n) = \frac{n}{\log_2 n} \tag{9}$$

$$C(n) = \frac{c(n)}{b(n)} \tag{10}$$

De acordo com a conclusão do artigo, há uma forte correlação entre a complexidade de Kolmogorov e os sinais de EEG. O valor da complexidade de Kolmogorov de acordo com os sinais da EEG inclinaram-se a diminuir conforme o aumento da fadiga mental. A Figura 3 ilustra a mudança dos valores da complexidade de Kolmogorov ao longo do tempo sob diferentes estados de fadiga mental.

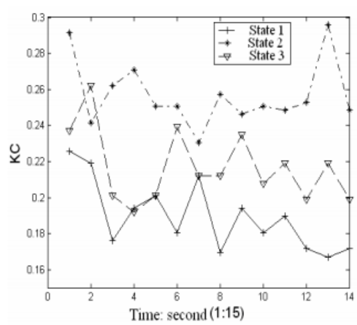

Figure 3: KC(Complexidade de Kolmogorov) ao longo do tempo.

5.4. Existem infinitos números primos

Prova: Supondo que não: Existem k primos $p_1, p_2, ..., p_k$ para algum $k \in \mathbb{N}$.

Assim, podemos pegar qualquer número $m \in \mathbb{N}$ e escrevê-lo como um produto desses k números primos:

$$m = {p_1}^{e}{}_1 ... {p_k}^{e}{}_k \tag{11}$$

Seja m c-incompressível e ter tamanho n. Pode-se descrever m como $e_1, ..., e_k$, afirmando que isso fornece uma descrição breve de m. Logo,

1. $e_i \leq \log m$
2. $|e_i| \leq \log \log m$
3. $|\langle e_1, ..., e_k \rangle| \leq 2k \log \log m$
4. $C(m) \leq 2k \log(n+1) + c$

Para um n suficientemente grande, isso contradiz $C(m) \geq n$, que decorre do fato de m ser aleatório.

Agora, sendo p_m o m-ésimo primo, a complexidade de Kolmogorov nos permite colocar um limite nesse valor [2]. Seja p_i o maior primo que divida m, podemos descrever m especificando p_i e $\frac{m}{p_i}$. Para m aleatório, temos:

$$C(m) \leq C(\langle i, \frac{m}{p_i} \rangle) \tag{12}$$

$$\leq 2 \log |i| + |i| + |\frac{m}{p_i}| \tag{13}$$

então:

$$\log m \leq 2 \log \log i + \log i + \log m - \log p_i \tag{14}$$

$$p_i \leq i(\log i)^2 \tag{15}$$

O teorema clássico descrito em [23] propõe que o i-ésimo primo está abaixo de $i \log i$, o que é significativamente perto [14].

5.5. Relação entre a Complexidade de Kolmogorov e Navalha de Occam

A navalha de Occam afirma que "A explicação mais simples é a melhor" [24]. A complexidade de Kolmogorov visa encontrar o programa de menor tamanho que produz uma *string* "s" [5]. Esse programa de menor tamanho é substancialmente o programa mais simples. Dessa forma, com base nas duas definições, pode-se aplicar a complexidade de Kolmogorov como uma formalização da Navalha de Occam [16].

6. Importância

A complexidade de Kolmogorov é um conceito relevante com muitas aplicações importantes. Conforme demonstrado na Seção 5, a complexidade é usada ilimitadamente, pois pode realmente ser aplicada a qualquer coisa que possa ser descrita como uma *string*.

Sua importância abrange não somente a categoria de ser usada como uma forma para aproximar (teoricamente) a dificuldade de uma função. É possível aplicar a complexidade de Kolmogorov em variadas áreas, como:

- Matemática: teoria da probabilidade e lógica.

- Física: Teoria do Caos e Termodinâmica.
- Ciência da Computação: Mineração de Dados, método da incompressibilidade, Caso médio *Heapsort*, Caso médio *Shellsort*.
- Filosofia e Biologia - aleatoriedade, inferência, sistemas complexos e similaridade de sequência.
- Além de muitos outros casos.

7. Considerações Finais

A complexidade de Kolmogorov e algumas de suas aplicações foram elucidadas nesse artigo na forma de pesquisa bibliográfica. A noção de Kolmogorov-Solomonoff-Chaitin apresenta uma visão diferente da informação e aleatoriedade que outros métodos da teoria da informação e aleatoriedade. O seu destaque está na ilimitabilidade de aplicações, pois pode ser aplicado a qualquer coisa que possa ser escrita como uma *string*. Novas pesquisas continuam, especialmente na área de complexidade de Kolmogorov Quantum e no ramo da inteligência artificial.

References

[1] H. R. Pagels, Os Sonhos da Razão, Editora Gradiva, 1990.

[2] M. Li, P. M. Vitnyi, An Introduction to Kolmogorov Complexity and Its Applications, 3 ed., Springer Publishing Company, Incorporated, 2008.

[3] A. Gammerman, V. Vovk, Kolmogorov complexity: Sources, theory and applications., Computer Journal 42 (1999).

[4] P. Grunwald, P. Vitányi, Shannon information and kolmogorov complexity, arXiv preprint cs/0410002 (2004).

[5] A. N. Kolmogorov, Three approaches to the quantitative definition of information, International Journal of Computer Mathematics 2 (1968) 157–168. doi:10.1080/00207166808803030.

[6] R. Fox, Roger hahn, pierre simon laplace 1749–1827:a determined scientist. cambridge, ma and london: Harvard university press, 2005. pp. xii+310. isbn 0-674-01892-3, British Journal for The History of Science 40 (2007). doi:10.1017/S0007087407000489.

[7] R. v. Mises, Grundlagen der wahrscheinlichkeitsrechnung, Mathematische Zeitschrift 5 (1919) 52–99.

[8] A. Wald, On cumulative sums of random variables, Ann. Math. Statist. 15 (1944) 283–296. URL: `https://doi.org/10.1214/aoms/1177731235`. doi:10.1214/aoms/1177731235.

[9] R. Gandy, Church's thesis and principles for mechanisms, in: J. Barwise, H. J. Keisler, K. Kunen (Eds.), The Kleene Symposium, volume 101 of *Studies in Logic and the Foundations of Mathematics*, Elsevier, 1980, pp. 123 – 148. doi:https://doi.org/10.1016/S0049-237X(08)71257-6.

[10] C. E. Shannon, W. Weaver, The Mathematical Theory of Communication, by CE Shannon (and Recent Contributions to the Mathematical Theory of Communication), W. Weaver, University of illinois Press, 1949.

[11] R. Fabbri, F. G. D. León, A statistical distance derived from the kolmogorov-smirnov test: specification, reference measures (benchmarks) and example uses, 2017. `arXiv:1711.00761`.

[12] G. J. Chaitin, Algorithmic information theory, IBM journal of research and development 21 (1977) 350–359.

[13] G. Chaitin, Information-theoretic computation complexity, IEEE Transactions on Information Theory 20 (1974) 10–15.

[14] A. Shen, V. A. Uspensky, N. Vereshchagin, Kolmogorov complexity and algorithmic randomness, volume 220, American Mathematical Soc., 2017.

[15] R. J. Solomonoff, The discovery of algorithmic probability, Journal of Computer and System Sciences 55 (1997) 73–88.

[16] T. Cover, J. Thomas, Elements of Information Theory, Wiley, 2012. URL: `https://books.google.com.br/books?id=VWq5GG6ycxMC`.

[17] A. M. Turing, On Computable Numbers, with an Application to the Entscheidungsproblem, Proceedings of the London Mathematical Society s2-42 (1937) 230–265. doi:10.1112/plms/s2-42.1.230.

[18] V. Nannen, A short introduction to kolmogorov complexity, 2010. arXiv:1005.2400.

[19] S. Evans, S. F. Bush, J. Hershey, Information assurance through kolmogorov complexity, in: Proceedings DARPA Information Survivability Conference and Exposition II. DISCEX'01, volume 2, IEEE, 2001, pp. 322–331.

[20] L. M. Spracklin, L. V. Saxton, Filtering spam using kolmogorov complexity estimates, in: 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07), volume 1, 2007, pp. 321–328.

[21] L. Zhang, C. Zheng, Analysis of kolmogorov complexity in spontaneous eeg signal and it's application to assessment of mental fatigue, in: 2008 2nd International Conference on Bioinformatics and Biomedical Engineering, 2008, pp. 2192–2194.

[22] A. Lempel, J. Ziv, On the complexity of finite sequences, IEEE Transactions on Information Theory 22 (1976) 75–81.

[23] J. Williamson, The Elements of Euclid, with Dissertations ..., The Elements of Euclid, with Dissertations, Clarendon Press, 1781. URL: https://books.google.com.br/books?id=tGP_ugEACAAJ.

[24] S. C. Tornay, Ockham: Studies and selections (1938).