



**Universidad del  
Rosario**

---

**Teoría de la información**

---

Realizado por  
Esteban Hernández Ramírez  
CC: 1'007.658.073

Supervisado por  
PhD. Carlos Eduardo Álvarez Cabrera

En el curso  
Opción de grado 1

Trabajo presentado a  
Comité de evaluación de opción de grado

Escuela de Ingeniería, Ciencia y Tecnología  
Matemáticas Aplicadas y Ciencias de la Computación  
Universidad del Rosario  
Monday 14<sup>th</sup> March, 2022

Bogotá-Colombia

## Introducción

La idea de una máquina de Turing que retorna una secuencia infinita de 1's y 0's, luego de haber operado sobre ella bajo reglas codificadas en otra secuencia de 1's y 0's, nos lleva a la pregunta de si dada una secuencia cualquiera, es posible determinar a partir de ella la secuencia de reglas codificadas que la generaron: más importante aún, cómo determinar la secuencia de reglas codificadas más corta capaz de generar la secuencia dada. De hecho, este problema ha sido formulado rigurosamente en la teoría de la computación como la Complejidad de Kolmogórov; introducida independientemente en la década de los 60's por Solomonoff ([6]), Kolmogorov ([3]), y Chaitin ([2]).

Se sabe que si la forma más sucinta de expresar la cadena fuera ella misma, entonces la cadena habrá sido generada de manera completamente aleatoria ([4]). También se sabe que, de manera general, la tarea de determinar la secuencia de reglas codificadas más corta es incomputable ([8]). No obstante, los esfuerzos de la comunidad científica se han centrado últimamente en idear formas de comprimir cualquier secuencia, lo más que sea posible. El trabajo pionero de Lempel y Ziv ([9]) demostró que existen formatos de compresión para los cuales el tamaño óptimo de un archivo comprimido tiende a estimar la entropía de la fuente generadora. Esta "entropía" fué una medida propuesta por Claude Shannon en la década de los 50's ([5]): esta medida es continua y es máxima cuando la probabilidad de generar cualquier símbolo en la secuencia es la misma. Lo anterior ha permitido calcular una aproximación a la Complejidad de Kolmogorov de una secuencia. Otras aproximaciones como [7] confirmarían más adelante lo visto por Lempel y Ziv, y acabaría por reforzar la creencia de que la Complejidad de Kolmogórov y la Entropía de Shannon están más relacionadas de lo que podría parecer en un principio.

Lo anterior posibilita diferenciar las reglas (aún sin conocerlas del todo) que generan algunas secuencias, usando los patrones que puedan presentarse en las propias secuencias para comprimirlas. Esto se ejemplifica perfectamente para el caso de textos (secuencias) escritos en algún lenguaje natural en diferentes idiomas: midiendo la entropía de sus fuentes mediante la compresión de las cadenas se puede distinguir un lenguaje del otro. La idea intuitiva es la siguiente: en este contexto, la entropía relativa se interpreta como el número de caracteres desperdiciados para codificar una cadena generada por una fuente  $\mathcal{B}$  mediante la codificación óptima para las cadenas generadas por otra fuente  $\mathcal{A}$  [1]. No en vano, para aproximar la entropía se pueden emplear algoritmos de compresión como lo demostraron Lempel y Ziv en ([9]). Y en ese orden de ideas, la entropía relativa por caracter entre las fuentes  $\mathcal{A}$  y  $\mathcal{B}$ ,  $S_{AB}$ , puede aproximarse como:

$$S_{AB} = \frac{\Delta_{Ab} - \Delta_{Bb}}{|b|}, \quad (1)$$

tal cómo se definió en [1]: Siendo  $\Delta_{AB} = L_{A+B} - L_A$  con  $L_X$  la longitud en bits del archivo comprimido  $X$ .

En realidad, es estos casos lo que nos interesará es estimar la entropía relativa entre las dos fuentes por medio de algoritmos de compresión, que es lo que describíamos intuitivamente en el parrafo anterior: ya que la entropía relativa se puede utilizar como una medida de distancia entre las cadenas generadas por dos fuentes distintas  $\mathcal{A}$  y  $\mathcal{B}$ . Es decir, qué tan bien funcionan las reglas que emplea la fuente  $\mathcal{A}$  para generar mensajes propios de la fuente  $\mathcal{B}$ , lo que nos dirá qué tan distintos son los mensajes que genera cada una.

En ese orden de ideas, en el presente escrito nos enfocamos en estudiar el problema de estimar la entropía relativa de dos fuentes generadoras de secuencias, al comprimir textos escritos en diferentes idiomas y detectar los cambios en el tamaño de los textos comprimidos. La forma en la que se organiza el texto a continuación es la siguiente: 1) cómo calcular la entropía relativa para textos escritos en lenguaje natural. 2) cómo detectar los cambios entre una secuencia y otra. 3) cómo interpretar estos cambios en términos de distancias filogenéticas del lenguaje. 4) Hacer una revisión de la aplicación de estos principios al análisis del ADN. 5) Implementación del algoritmo para buscar distancias entre algunas secuencias de ADN.

## La relación entre la entropía relativa y la compresión de cadenas

En esta sección, nos concentramos en explicar la relación que existe entre la entropía relativa y la codificación de cadenas, específicamente en cómo el problema inicial se reduce a un tipo muy particular de esto último: la compresión de cadenas.

### La entropía relativa funciona como una medida de comparación entre cadenas

Para saber qué tan bien funcionan las reglas que emplea una fuente  $\mathcal{A}$  para generar mensajes propios de otra fuente  $\mathcal{B}$ , lo que nos dirá qué tan distintos son los mensajes que genera cada una, se puede utilizar el número de caracteres desperdiciados para codificar una secuencia emitida por la fuente  $\mathcal{B}$  con el código óptimo para la fuente  $\mathcal{A}$ : la entropía relativa entre las fuentes  $\mathcal{A}$  y  $\mathcal{B}$  (Revisar Kullback-Leibler [13]).

Lo anterior se justifica de la siguiente manera, suponga que se tienen dos fuentes ergódicas  $\mathcal{A}$  y  $\mathcal{B}$  que están emitiendo secuencias de '0' y '1':  $\mathcal{A}$  emitiendo '0' con probabilidad  $p$  y '1' con probabilidad  $1 - p$ , mientras que  $\mathcal{B}$  emite un '0' con probabilidad  $q$  y un '1' con probabilidad  $1 - q$ . La codificación óptima de las secuencias de la fuente  $\mathcal{A}$  consiste en emplear  $\log_2\left(\frac{1}{p}\right)$  bits para codificar el '0' y  $\log_2\left(\frac{1}{1-p}\right)$  para codificar el '1', mientras que para la fuente  $\mathcal{B}$ , la codificación óptima emplea  $\log_2\left(\frac{1}{q}\right)$  bits para codificar el '0' y  $\log_2\left(\frac{1}{1-q}\right)$  para codificar el '1'. De esta manera se estará empleando la menor cantidad de bits posible para codificar los mensajes de ambas fuentes: llámese la entropía de Shannon, tal como se definió en [5].

En ese orden de ideas, la entropía por caracter de una secuencia emitida por  $\mathcal{B}$  en la codificación óptima para  $\mathcal{A}$  puede obtenerse a partir de una secuencia infinitamente larga de la fuente  $\mathcal{B}$  que ha sido codificada con las cantidades de bits de la fuente  $\mathcal{A}$ , de la siguiente manera: considere que  $\{X\}_{i=0}^{\infty}$  es una muestra aleatoria infinitamente grande de variables aleatorias independientes de Bernoulli con media  $\mu = q$ , entonces

$$\lim_{N \rightarrow \infty} \frac{1}{N} \cdot \left[ \log_2\left(\frac{1}{p}\right) \sum_{i=0}^N X_i + \log_2\left(\frac{1}{1-p}\right) \left(N - \sum_{i=0}^N X_i\right) \right] = q \cdot \log_2\left(\frac{1}{p}\right) + (1-q) \cdot \log_2\left(\frac{1}{1-p}\right)$$

por el teorema del límite central. No obstante, las hipótesis de convergencia e independencia del teorema están garantizadas bajo la asunción de que  $\mathcal{A}$  y  $\mathcal{B}$  son fuentes ergódicas. Así, la entropía por caracter de una secuencia emitida por  $\mathcal{B}$  en la codificación óptima para  $\mathcal{B}$  es  $q \cdot \log_2\left(\frac{1}{q}\right) + (1-q) \log_2\left(\frac{1}{1-q}\right)$ . Análogamente para una secuencia emitida por  $\mathcal{A}$ .

De esta manera, el número de bits por caracter desperdiciados para codificar una secuencia emitida por  $\mathcal{B}$  con la codificación óptima para  $\mathcal{A}$  se puede calcular como la diferencia:

$$\begin{aligned} S_{AB} &= \left[ q \cdot \log_2\left(\frac{1}{p}\right) + (1-q) \log_2\left(\frac{1}{1-p}\right) \right] - \left[ q \cdot \log_2\left(\frac{1}{q}\right) + (1-q) \log_2\left(\frac{1}{1-q}\right) \right] \\ &= \left[ q \cdot \log_2\left(\frac{1}{p}\right) - q \cdot \log_2\left(\frac{1}{q}\right) \right] + \left[ (1-q) \log_2\left(\frac{1}{1-p}\right) - (1-q) \log_2\left(\frac{1}{1-q}\right) \right] \\ &= q \left[ \log_2\left(\frac{1}{p}\right) - \log_2\left(\frac{1}{q}\right) \right] + (1-q) \left[ \log_2\left(\frac{1}{1-p}\right) - \log_2\left(\frac{1}{1-q}\right) \right] \\ &= q \left[ \log_2\left(\frac{q}{p}\right) \right] + (1-q) \left[ \log_2\left(\frac{1-q}{1-p}\right) \right] \\ &= -q \cdot \log_2\left(\frac{p}{q}\right) - (1-q) \cdot \log_2\left(\frac{1-p}{1-q}\right), \end{aligned}$$

la entropía relativa de  $\mathcal{A}$  y  $\mathcal{B}$ . De manera inversa se define la entropía relativa de  $\mathcal{B}$  y  $\mathcal{A}$ :  $S_{BA} = -S_{AB}$ .

En conclusión, midiendo la entropía relativa podemos saber qué tan distintos son los mensajes que generan dos fuentes. Así que, hemos reducido (y restringido) el problema original a medir la entropía relativa de dos fuentes. Pero entonces queda una pregunta: ¿cómo calcular esta entropía relativa?. Necesitamos un procedimiento que estime la entropía relativa de un par de cadenas de entrada. Si bien existen varias maneras de calcular la entropía relativa de dos cadenas [see for example 10,14], la que interesa a este escrito es la de calcular la entropía relativa por medio de la compresión de cadenas. Antes, conviene presentar el formato de codificación LZ77 y una revisión de los resultados formales que respaldan esta aproximación [9].

## Formato de codificación LZ77

En esta sección, explicamos en qué consiste esta codificación y cómo se obtiene a partir de una cadena de entrada.

## Entropía relativa en términos del tamaño de archivos comprimidos

Ahora, procedemos a derivar una cota superior para el ratio de compresión obtenible por el algoritmo LZ77 que presentamos antes. Esto lo hacemos para el peor de los casos: un mensaje de la fuente de longitud  $(n - L_s)$ , donde  $n$  es la longitud del buffer y  $L_s$  es la longitud máxima de la palabra. Al final, notaremos que la cota para este caso, también aplica para mensajes de longitud exactamente igual que  $n - L_s$  o mayor, gracias a la forma (y generalidad) de la cota que se obtiene.

### Análisis de redimiento

Por cómo se definió, sabemos que toda cadena  $S \in \sigma\{n - L_s\}$  se descompone en una colección de subcadenas,  $\{S_i\}_{i=0}^{N(S)}$ , de longitudes máximas  $L_s$  (variables) durante el procedimiento de codificación LZ77, antes de mapear cada una de estas a su respectiva palabra código de longitud fija  $L_c$ . Asociado a cada una de estas asignaciones está una cantidad continua y positiva, que da cuenta de qué tanto la codificación está comprimiendo:

$$\frac{L_c}{\ell(S_i)}.$$

Esta cantidad es estrictamente mayor que 1 si y solo si la longitud de la palabra código es estrictamente mayor que la palabra codificada. Así, cuando  $L_c/\ell(S_i) < 1$  diremos que la codificación está comprimiendo la subcadena  $S_i$ . En este orden de ideas, también podemos saber, de manera global, si la codificación está comprimiendo la palabra  $S$ , verificando si

$$\rho(S) = \frac{L_c \cdot N(S)}{\left(\sum_{i=1}^{N(S)} \ell(S_i)\right)} = \frac{L_c}{n - L_s} \cdot N(S) < 1.$$

No en vano, este concepto posibilita definir un criterio para evaluar el rendimiento de la codificación LZ77 sobre una fuente en particular:

$$\rho(\sigma) = \frac{L_c}{n - L_s} \cdot N \quad \text{con} \quad N = \max_{S \in \sigma\{n - L_s\}} N(S),$$

la "tasa de compresión máxima alcanzable" por el algoritmo para la fuente  $\sigma$ . Observe que en este caso (el peor de los casos), esta tasa es óptima, pues es la tasa mínima alcanzable para cualquier longitud de palabra, al tiempo que es la máxima entre las palabras de longitud  $n - L_s$ . Como tál, merece la pena estudiar este valor y saber hasta qué punto puede mejorarse, o en otras palabras, acotarse superiormente.

Para eso, las cantidades  $L_c$  y  $(n - L_s)$  son fijas, únicamente  $N$  puede variar. Por lo tanto, utilizaremos a  $N$  para inducir esa cota superior: aumentando el valor de  $N$ , como  $N'$ , y expresando a  $N'$  de una manera que pueda ajustarse. Antes debemos expresar a  $N'$  en términos que permitan ajustar la cota.

Sea  $Q \in \sigma\{n - L_s\}$  tal que  $N(Q)$ , el número de subcadenas en las cuales  $Q$  es descompuesto por el algoritmo, es igual a

$$N = \max_{S \in \sigma\{n - L_s\}} N(S).$$

De modo que, el algoritmo descompuso a  $Q$  como

$$Q = Q_1 Q_2 \cdots Q_N.$$

Ahora bien, el paso número dos del algoritmo LZ77 garantiza que si  $\ell(Q_i) = \ell(Q_j)$  y  $\ell(Q_j) < L_s$ , para algunos  $i < j < N$ , entonces  $Q_i \neq Q_j$ . Es decir, cualesquiera de estos  $Q'_k$ s, llamensen  $Q_i$  y  $Q_j$ , pudieran tener la misma longitud, aunque no serán el mismo bloque, es decir  $i \neq j$  ( $Q_i \neq Q_j$ ).

Esto último nos permite saber que,  $K_m$ , el número de  $Q'_i$ s de tamaño  $m$  ( $1 \leq m \leq L_s$ ), está bien definido para todo  $i$  ( $1 \leq i \leq N - 1$ ), ya que cada  $Q_i$  estará contando una única vez dentro de su  $K_m$ . Ahora, podemos expresar las cantidades  $N$  y  $(n - L_s)$  en términos de los  $K_m$  como:

$$N = 1 + \sum_{m=1}^{L_s} K_m,$$

y

$$n - L_s = \ell(Q_N) + \sum_{m=1}^{L_s} m \cdot K_m.$$

No obstante,  $(n - L_s)$  está fijo, así que se pueden sobrestimar los valores de  $K_m$  para  $1 \leq m \leq (L_s - 1)$ , a expensas de  $K_{L_s}$ , únicamente modificando a  $N$ . No en vano, los valores de  $K_m$  pueden ser acotados convenientemente para definir la forma y el ajuste de la cota.

Particularmente, sabemos que

$$K_m < \sigma(m) \quad \text{para } 1 \leq m \leq L_s - 1.$$

## References

- [1] Dario Benedetto, Emanuele Caglioti, and Vittorio Loreto. “Language Trees and Zipping”. In: Physical Review Letters 88.4 (2002). DOI: 10.1103/PhysRevLett.88.048702.
- [2] G. J. Chaitin. “A theory of program size formally identical to information theory”. In: Journal of the ACM 22.3 (1975), pp. 329–340. DOI: 10.1145/321892.321894.
- [3] A. N. Kolmogorov. “Three approaches to the quantitative definition of information”. In: Problems in Information Transmission 1.1 (1965), pp. 1–7.
- [4] Toshiko Matsumoto. “Biological Sequence Compression Algorithms”. In: Genome Informatics 11 (2000), pp. 43–52. DOI: 10.11234/GI1990.11.43.
- [5] C. E. Shannon. “A mathematical theory of communication”. In: The Bell System Technical Journal 27.3 (1948), pp. 379–423. DOI: 10.1002/j.1538-7305.1948.tb01338.x.
- [6] R. J. Solomonoff. “A formal theory of inductive inference”. In: Information and Control 7 Parts 1 and 2.1-22 (1964), pp. 224–254. DOI: [https://doi.org/10.1016/S0019-9958\(64\)90223-2](https://doi.org/10.1016/S0019-9958(64)90223-2).
- [7] Andreia Teixeira et al. “Entropy Measures vs. Kolmogorov Complexity”. In: Entropy 13.3 (2011), pp. 595–611. ISSN: 1099-4300. DOI: 10.3390/e13030595. URL: <https://www.mdpi.com/1099-4300/13/3/595>.
- [8] Paul M.B. Vitányi. “How Incomputable Is Kolmogorov Complexity?”. In: Entropy 22.4 (2020). ISSN: 1099-4300. DOI: 10.3390/e22040408. URL: <https://www.mdpi.com/1099-4300/22/4/408>.
- [9] Jacob Ziv and Abraham Lempel. “A Universal Algorithm for Sequential Data Compression”. In: IEEE TRANSACTIONS ON INFORMATION THEORY 23.3 (1977), pp. 337–343. DOI: 10.1109/TIT.1977.1055714.

Participantes:

Firma: \_\_\_\_\_  
Álvarez Cabrera, Carlos Eduardo. Ph.D.  
Tutor, Matemáticas Aplicadas y Ciencias de la Computación  
Tuesday 18<sup>th</sup> May, 2021

Firma: \_\_\_\_\_  
Hernández Ramírez, Esteban.  
Estudiante, Matemáticas Aplicadas y Ciencias de la Computación  
Tuesday 18<sup>th</sup> May, 2021