Análisis iterativo en juegos de información. Formalización y solución algorítmica de *Wordle*.

Sofía Castaño[1], **Alfonso C. Fiero**[2], **Jerónimo Osorio**[3], and **Valentina Ortega** [4]

[1-4]Universidad EAFIT

Resumen

El artículo se centra en develar las particulares y muy fuertes relaciones matemáticas que Wordle comparte con la teoría de la probabilidad, en específico, con la teoría de la información. Bajo el más cuidadoso rigor matemático se desarrolla una completa abstracción del juego y de las acciones que se pueden realizar dentro de este como un sistema construido con básicas operaciones de conjuntos y con muchas definiciones originales. En general puede notarse que la intención de los autores es proveer de una justificación férrea la aplicabilidad de la teoría de la información que el divulgador matemático Grant Sanderson descubrió dentro del juego, al mismo tiempo se busca superarle en los resultados de su solución algorítmica. Como Grant, el gran propósito es encontrar una lista de palabras que sirvan como apertura estratégica para el juador humano y desarrollar un autómata capaz de ganar en la menor cantidad de turnos posibles.

Palabras clave: Wordle; Patrón; Evento; Variable aleatoria; Teoría de información; Probabilidad; Entropía.

Abstract

This article revolves around the strong and particular mathematical bonds that *Wordle* shares with the theory of probabilities, specifically with information theory. It is under the most rigorous mathematical care that we can develop an abstraction over the game and the actions that are allowed in it as if it was a system built in with basic set operations and with various original definitions. As an overview we can note that the intentions of the authors are focused in providing a tenacious justification in the applicability of information theory that the mathematical divulgator Grant Sanderson discovered inside this game, at the same time that is looked forward the possibility to overcome his results with the algorithmic solution. As Grant pretends, the ultimate purpose relies in finding a list of words that serve as an opening strategy for the human player and develop a automaton capable of winning in the least possible amount of attempts. .

Keywords: Wordle; Hints; Pattern; Event; Random variable Information theory; Probability; Entropy.

Introducción

Durante los meses de febrero y marzo surgió como fenómeno mediático el juego diario de acertijos lexicográficos *Wordle*. Este pasatiempo consta de adivinar una palabra de 5 letras en 6 intentos. Por cada intento se debe ingresar un termino reconocido en el diccionario y se obtiene mediante un código de color información sobre la existencia y la posición que ocupan las letras que le componen. Existen 3 marcas de color. Si la letra se tintura de forma gris, la solución no la contiene; Si se tintura de forma amarilla, la solución la contiene en una posición distinta; Y si se tintura de verde, la letra se encuentra en la posición correcta.

A partir de esto surgio un mundo de posibilidades de analisis, dentro de las cuales se crea nuestro proyecto, una interesante oportunidad para abordar un tema aparantemente simple pero que en realidad resguarda una complejidad fascinante.

1. Sencillez

Desde una aproximación ingenua, el ingreso de cada nueva palabra se pueden cosiderar un ensayo de bernoulli en el que se tiene una probabilidad p^* de concluir el juego. Construyendo una primer solución con fuerza bruta podemos considerar una única *gran partida* deslimitada de intentos.
Asumimos que el juego se contextualiza dentro de un lenguaje finito $\mathfrak{L}$ que posee un número N de palabras y un número n de palabras admisibles. La palabra de apertura siempre resulta con una probabilidad $\frac{1}{n}$ de ser la correcta porque en principio se está realizando una honesta adivinanza dentro de todo el espacio del leguaje. La condición inicial puede provocar que el juego sea infinito porque dado un primer fracaso (caso con probabilidad $1-\frac{1}{n}$) y una opción por este objeto incorrecto, perderemos de manera indefinida.
Ignorando deliberadamente las pistas proveídas por los códigos de color, un jugador racional procedería en el juego descartando siempre las palabras que haya usado en turnos anteriores. Si movemos nuestro juego en un espacio probabilístico $\mathfrak{P}$ cuyo espacio muestral (ω) y su $\sigma-$álgebra $(\mathfrak{F})$ son respectivamente el conjunto de palabras elegibles en el lenguaje $\mathfrak{L}$ y el conjunto potencia de este, es posible plantear una medida de probabilidad indicadora a estos conjuntos dependiendo de si la palabra *ganadora* $(\star)$ es contenida en el conjunto: Sea x un elemento en $\mathfrak{F}$, definimos f como sigue:

$$f(x) = \begin{cases} 1 & \text{si } \star \in x \\ 0 & \text{si } \star \notin x \end{cases} \tag{1}$$

En un lenguaje con n palabras admisibles es posible formalizar una variable aleatoria X que determine el número de intentos hasta terminar el juego asociando el cardinal de los elementos de $\mathfrak{F}$ que contienen a $\star$ con los números naturales. Esto porque el registro de un juego concluido puede verse como los elementos del conjunto potencia que incluyen a $\star$. Bajo las condiciones mínimas, en una partida ganada no son relevantes ni el orden ni la identificación de las palabras incorrectas ingresadas. La manera de demostrar que X cumple las condiciones de variable aleatoria es asignando a los intervalos $(-\infty, i]$ con $i \in \mathbb{N}$ un elemento en $x \in \mathfrak{F}$ tal que su cardinalidad corresponda con i, i.e. llamando w_i a los posibles arreglos de todas las posibles longitudes de palabras distintas a $\star$:

$$\begin{gathered} X^{-1}\ (-\infty, 0] = \phi \\ X^{-1}\ (-\infty, 1] = \{\ \star\ \} \\ X^{-1}\ (-\infty, 2] = \{\ w_1, \star\ \} \\ \ldots \\ X^{-1}\ (-\infty, k] = \{\ w_k, \star\ \} = \omega \end{gathered}$$

Es natural notar que hay un número finito k de conjuntos x que cumplen $\star \in x$ por lo que para cualquier otro $i > k$ el boreliano $(-\infty, i]$ tiene como imagen inversa a $\{\ w_k, \star\ \}$, ahora es necesario inducir una medida de probabilidad que escale de manera sensata la función indicadora definida en $\mathfrak{P}$. Planteemos g_t como el evento en que el juego termina satisfactoriamente en t intentos en un lenguaje con n palabras admisibles, luego la probabilidad del evento g_t, $P(g_t)$, es equivalente a la probabilidad $P(X = t)$. los casos base ocurren cuando $t = 0$ y $t = 1$, es decir, la probabilidad de ganar sin jugar y después de jugar por primera vez.

$$P(g_0) = 0$$

$$P(X=1)=P(g_1)=\tfrac{1}{n}$$

Extendiendo esta proposición, para la comprobación de un juego finito con t turnos es necesario ganar en t y haber fracasado en los $t-1$ turnos anteriores y, fracasar en $t-1$ turnos, es suficiente para determinar el turno t es una victoria. Podemos considerar este razonamiento como una justificación para asumir cada turno como un evento independiente dentro del espacio $\mathfrak{P}$

$$P(g_t)=P(g_t|\neg g_{t-1}) \implies P(g_t\cap\neg g_{t-1})=P(g_t)P(\neg g_{t-1})$$

Esta condición de independencia nos permite plantear un patrón para las probabilidades de los primeros eventos

$$P(X=1)=\frac{1}{n}$$
$$P(X=2)=P(g_2)P(\neg g_1)=\frac{1}{n-1}\left(1-\frac{1}{n}\right)=\left(\frac{1}{n-1}\right)\left(\frac{n-1}{n}\right)=\frac{1}{n}$$
$$P(X=3)=P(g_3)P(\neg g_2)=\left(\frac{1}{n-1}\right)\left(\frac{n-1}{n}\right)=\frac{1}{n}$$
$$\ldots$$

De manera general
$$P(X=t)=P(g_t)P(\cap_{j=0}^{t-1}\neg g_{j-1}) \tag{2}$$

queremos demostrar que sin importar el valor de t, $(1\leq t\leq n)$, la probabilidad permanece constante: Expandiendo la expresión (2) hasta el caso extremo en que se han agotado las $n-1$ palabras distintas de $\star$, es decir, concluyendo el juego en el turno n

$$P(X=n)=P(g_n)P(\neg g_{n-1})P(\neg g_{n-2})\ldots P(\neg g_1)$$
$$P(X=n)=\left(\frac{n-1}{n}\right)\left(\frac{n-2}{n-1}\right)\left(\frac{n-3}{n-2}\right)\cdots\left(\frac{1}{2}\right)$$
$$=\prod_{i=0}^{n-2}\frac{n-1-i}{n-i}=\frac{(n-1)!}{n!}=\frac{1}{n} \tag{3}$$

Para un turno $k\geq 2$ se cumple la siguiente expresión:

$$P(X=k)=\left(\frac{n-1}{n}\right)\left(\frac{n-2}{n-1}\right)\left(\frac{n-3}{n-2}\right)\cdots\left(\frac{1}{n-(k-1)}\right)$$
$$P(X=k)=\frac{1}{n-(k-1)}\prod_{i=1}^{k-1}\frac{n-i}{n-(i-1)}$$

No es difícil comprobar que esta productoria converge continuamente a $\frac{1}{n}$ y el caso más extremo fue demostrado en la expresión (3). Luego, la densidad para la distribución de esta variable aleatoria es uniforme dentro de el conjunto $\{1,2,\ldots,n\}$, ie.

$$f_X(x)=\begin{cases}1/n & \text{si } x=1,2,3\ldots,n\\ 0 & \text{de otro modo}\end{cases} \tag{4}$$

Este resultado puede interpretarse como la falta de memoria que las condiciones mínimas proveen para el desarrollo del juego. Esto es porque por simpleza estamos tratando a *Wordle* como si fuese una adivinanza (o lotería) en lugar de reconocer su estructura orgánica como rompecabezas. Una estrategia de prueba y error es fundamentalmente desesperanzadora si intentamos ganar en 6 intentos o menos pues

$$P(X \leq 6) = 6/n$$

$$E(X) = \sum_{i=1}^{n} \frac{i}{n} = \frac{1}{n} \sum_{i=1}^{n} i = \left(\frac{1}{n}\right) \frac{n(n+1)}{2} = \frac{n+1}{2}$$

Ejemplificando, al tomar el inglés como nuestro lenguaje $\mathfrak{L}$ reconocemos 8996 palabras con 5 letras, luego $|\omega| = 8996$, y por lo tanto la probabilidad de llevar el juego exitosamente en menos de seis intentos y la media de intentos hasta concluir el juego serían:

$$P(X \leq 6) = 6/8996 \approx 8{,}9(10^{-4})$$

$$E(X) = \sum_{i=1}^{n} \frac{i}{8996} = 4498{,}5$$

Si asignamos la estrategia de prueba y error dentro de un autómata, el algoritmo provee la ventaja de tener una implementación simple y un tiempo de ejecución muy bajo. La estructura general del procedimiento sería como se ve a continuación

```
#Procesar el espacio muestral
avaliable_words = read_table('set.txt')
#selecci n de palabra estrella
star = random.choice(avaliable_words)
turn=0
#simulaci n prueba y error
while(star != available_words[i]):
    game.attempt(available_words[i])
    turn += 1
```

Ignorando la estructura con la que se implemente la comprobación de los estados del juego, la solución a este requiere de un único ciclo simple que lleve cuenta de los turnos en que se descifra la palabra ($\star$).

2. Memoria

La manera en que se recompensa a los jugadores por sus equivocaciones es a través del código de color. Para la interpretación correcta de cada pista debe asumirse que el jugador (autómata) es omnisciente de su lenguaje y es continuamente capaz de recortar sus posibilidades hasta el punto en que solo reste intentar la palabra "$\star$". Esta estrategia de solución deshace por completo la condición de independencia y por lo tanto deja de ser válida la densidad propuesta en la expresión (3) para la v.a. X. Solo es posible rescatar que

$$P(g_0) = 0$$

$$P(g_1) = 1/n$$

$$P(g_t) = P(g_t | \cap_{j=0}^{t-1} \neg g_j)$$

En principio es una tarea muy ardua generar una expresión general para $P(X = t)$, pues solo hasta el turno 6 la palabra "$\star$" puede obtenerse en un estimado de $\frac{n!}{(n-6)!}$ escenarios (cada escenario con un resultado distinto). Puede ser que hasta después del quinto turno el espacio muestral se haya reducido en $n-1$ palabras, entonces es inmediato que $P(X = 6) = 1$, como también que solo se redujera en $n-500$, entonces $P(X = 6) = \frac{1}{500}$. Los ejemplos anteriores son necesarios para revisar que el evento ”ganar en el turno t”,no es cuantificable. Proponemos entonces el evento ”ganar en el turno t dado que se han intentado las palabras del arreglo y”

Ahora es necesario definir una variable auxiliar l_i que represente el conjunto de palabras descartadas en el turno i. Es inmediato notar que una misma palabra no puede ser descartada 2 veces y por lo tanto, los conjuntos de palabras descartadas son mutuamente excluyentes dentro del espacio muestral. Bajo estas condiciones intuimos que la función de densidad tiene la siguiente forma, pero no podemos demostrar su convergencia en los márgenes de este artículo pues se requeriría un análisis para cada combinación de conjuntos l_i.

$$P(X=t) = P(g_t|y) = \left(n - \sum_{i=0}^{t-1} |l_i|\right)^{-1} \tag{5}$$

Consideramos que en cada turno las palabras son equiprobables entre ellas, pero no son equiprobables entre turnos. Esto último hace que el problema rebote continuamente a depender de las reducciones en el espacio muestral. A continuación se describe esta reducción de manera algorítmica:

```
attempt = random.choice(available_words) #palabra estrella
color_chain = attempt.getColors() #patr n de colores
char = 0
for j in color_chain:
    switch(j):
        case G: #Letras grises
            for word in available_words: # Palabras lista
                if attempt[char] in word:
                    word.pop()
            #si la letra esta en la palabra,
            se descarta
        case V: #Letras verdes
            for word in avaliable_words:
                if attempt[char] != word[char]:
                    word.pop()
            #si la palabra no tiene la letra,
            se descarta
        Case A: #letras amarillas

            #frecuencia de la letra
            freq = letter.getFrequency(attempt)
            for word in avaliable_words:
                #Condici n, existencia y cardinal
                if  attempt[char] not in word[char]
                or
                letter.getFrequency(word) < freq:
                    word.pop()
            char+=1
```

El método **getColors()** debe retornar una cadena que provea la relación entre los carácteres de la palabra intentada y ”$\star$”. Esta se construye como una permutación generada por las iniciales de los colores descriptivos del juego de tal forma que haya una correspondencia carácter-color entre **attempt** y $\star$. El procesamiento del patrón de colores nos permite justificar que el orden en que se ingresan las palabras es irrelevante para el desarrollo del juego.

Considere una "$\star$"fija y una colección L de conjuntos l_i. se sabe que $\cap_L l_i = \phi$, luego

$$\sum_L |l_i| = \left| \bigcup_L l_i \right|$$

De esta forma es que para un mismo conjunto de errores la cantidad de palabras que se pueden descartar es invariante.
Con el resultado anterior nos podemos permitir conservar en totalidad el espacio $\mathfrak{P}$ en que se desarrolla el juego y nos aseguramos de que la variable aleatoria X esté bien definida. Hasta este punto hemos logrado una formalización matemática para *Wordle*, nos falta entonces plantear un proceso de optimización para solucionarlo. El principal problema que enfrentamos es el desarrollo de una estrategia para la elección de las palabras. Sabiendo que el objetivo siempre es maximizar la probabilidad de éxito o minimizar el espacio muestral, cada turno en que no se encuentre a "$\star$" se deben descartar la mayor cantidad de palabras posibles. Este criterio de optimización nos invita a implementar herramientas de teoría de la información, en específico, una medición de entropía.Esta estrategia de solución fue originalmente propuesta por Grant Sanderson.La siguiente sección es una formalización rigurosa a las ideas expuestas en su vídeo *"Solving Wordle using information theory"*.

3. Información

Podemos considerar un sistema total W que represente el estado del juego en un historial de t de ensayos S tal que .

$$W = \left(\bigcup_{i=1}^{t-1} \neg g_i \right) \cup g_t = \{\neg g_1, \neg g_2, ..., g_t\}$$

En este se cumple la condición necesaria en la que solo puede ocurrir un único evento de manera simultánea, estos eventos siendo g_t o $\neg g_t$. La respuesta inmediata del juego para diferenciar la ocurrencia de estos eventos es el despliegue de un código de color que ratifica las similitudes entre la palabra recién ingresada y "$\star$". Cada intento puede ser visto como una quíntupla ordenada en la que cada ínidice es la inicial del color con el que se cualifica a la letra el evento. Sea C el conjunto de las iniciales para los colores *Verde, Gris y Amarillo*

$$C = \{V, G, A\}$$

Cada intento produce un patrón de color dentro de las permutaciones posibles para los elementos de C, decimos entonces que este sistema de evaluación permite representar el evento g_t como aquel en que todas las letras se tinturan de color verde.

$$g_t \sim (V, V, V, V, V)$$

Y cualquier otro evento $\neg g_i$ sería encapsulado dentro del quinto producto cartesiano de C

$$\{\neg g_i\} \sim C^5 - \{(V, V, V, V, V)\}$$

Bajo estas relaciones surge una nueva manera de representar W a través de los elementos de C^5. Llamando c_i la quíntupla representativa para la palabra utilizada en el turno g_i podemos simplificar la representación del sistema tal que

$$W = \bigcup_{i=1}^{t} c_i$$

Cada elemento c_i puede considerarse un sub-evento de g_i pues es la consecuencia automática y complementaria para describir la información obtenida por cada palabra ingresada. Con las herramientas hasta ahora propuestas es que podemos definir W a partir de los tan mencionados turnos. Un turno T se define como algún elemento del producto cartesiano entre ω y C^5 y representa la palabra escogida y su patrón de color representativo. Queda así que un juego W de t turnos es formalmetne definido como

$$W = \bigcup_{i=0}^{t} T_i$$

sabiendo que el último turno será $T_t = (\star, (V,V,V,V,V))$
Es entonces que es posible proponer una medición que describa la efectividad de cada turno para aumentar la probabilidad de encontrar a $\star$, i.e. podemos medir la *información* generada en el sistema por cada turno. Es necesario aclarar que para medir correctamente la información, se está considerando la probabilidad de el turno T dentro del espacio muestral ω. La densidad para esta probabilidad se construye al responder la pregunta ¿Entre las palabras previamente disponibles, cuantas de ellas cumplen el patrón actualmente observado?

$$P(T = y) = \frac{n - \sum_{i=0}^{t} |l_i|}{n - \sum_{i=0}^{t-i} |l_{i-1}|} \qquad y = 1, 2, 3... \tag{6}$$

Podemos decir que la información, en bits, de intentar una palabra arbitraria w con patrón de color c en un turno y está dada por la expresión

$$I(T = y) = -log_2(P(T = y)) \tag{7}$$

Si queremos estimar la cantidad de información que nos ofrece una palabra antes de ingresarla, es decir, predecir el rango en que se recorta el espacio muestral, conviene fijar la palabra que se quiere ingresar y computar los 243 posibles patrones que pueden emerger de ella. Abusando de la notación, es viable fragmentar cada turno dependiendo del patrón de colores esperados, tanto así que $P(T = y_i)$ se lee como la probabilidad del turno y en que se presenta un patrón de colores i. El proceso de estimación es realizado a partir de la definición de *entropía de Shannon* como la esperanza de la información para un evento t, i.e.

$$H(T = y) = E(I(T = y)) = -\sum_{i=1}^{243} P(T = y_i) log_2(P(T = y_i)) \tag{8}$$

$H(T = y)$ se interpreta directamente como la media de bits de información que se pueden obtener en el turno y para una palabra w, siendo así, si computamos esta media para cada una de las palabras admisibles retantes en dicho turno, y escogemos aquella con mayor entropía, estaremos asegurando una notable, aunque no máxima, reducción del espacio muestral. Es natural ver que la información incrementa de manera exponencial cuando la probabilidad del turno es muy cercana a 0, esto implica que entre mayor sea la rareza de la palabra y su patrón de color dentro del espacio muestral, mayor información provee. Esta rareza puede ser vista desde los términos prácticos a través de la frecuencia de que tienen las palabras en contextos literarios. Para mejorar el desempeño de nuestra solución algorítmica proponemos implementar una caracterización a la entropía de cada palabra al dividirla por un estimado de la frecuencia absoluta dentro de todo el lenguaje. El proceso anterior es facilitado por el conjunto de datos *Frequency lists; Spoken and written english* proveído por la Universidad de Lancaster.

Este escalamiento está pensado para asimilar la noción humana que existe para describir aquello que es común. Una entropía muy alta implica la presencia de probabilidades muy bajas entre los patrones de color, esto implica posiciones y presencia de letras poco comunes en el lenguaje, lo que a su vez implica la presencia de palabras exóticas pero muy cargadas de información. Una entropía baja está atribuida a un comportamiento cuasi-uniforme para la probabilidad de los turnos a escoger y esto implica un constreñimiento poco significativo del espacio muestral,el algoritmo determina la palabra más frecuente entre las mejores candidatas y escoge aquella que arroje una menor relación entropía/frecuencia.

4. Resultados

Apertura	Entropía (bits)
banes	7.8298
braes	7.8888
bares	8.0583
cares	8.0691
bales	8.1068

Cuadro 1. Aperturas recomendadas respecto a su entropía

Apertura	Entropía (bits)
which	$4{,}51 \times 10^{-4}$
would	$1{,}10 \times 10^{-3}$
their	$1{,}39 \times 10^{-3}$
could	$1{,}92 \times 10^{-3}$
these	$2{,}88 \times 10^{-3}$

Cuadro 2. Aperturas recomendadas respecto a su razón entropía/frecuencia

Algoritmo Usado	media T.E. (s)	media de intentos	% derrotas
Sol. ingenua	$5{,}19 \times 10^{-5}$	4331	100
Sol. memoria	0,09	3.533	1.35
Sol. 3b1b (v3)	*	3.419	*

Cuadro 3. Comparacón de desempeño por tiempo de ejecución, media de intentos por juego y porcentaje de fracasos

A partir de la automatizacion de nuestro proceso se logró medir y ordenar la entropía inicial de todas las palabras en ω. Fue posible presentar una lista con las mejores 5 palabras de apertura para cada criterio de juego con respecto a sus bits esperados de información. Estas palabras son significativamente distintas a las encontradas bajo el método de grant por la manera en que se administra el espacio muestral dentro de cada algoritmo. Esto se da por la rigídez de nuestro método en la interpretación de patrones de color amarillo. Comparando las simulaciones de desempeño entre nuestro algoritmo de solución de memoria y la versión final del algoritmo diseñado por Grant Sanderson se muestra que no existe una diferencia significativa entre la media de intentos necesarios para finalizar el juego. Pero es necesario mencionar que Sanderson implementa un ajuste para limitar el espacio muestral a las 2315 palabras que son arbitrariamente escogidas y públicamente conocidas como soluciones del juego.

(3Blue1Brown, 2022) (ucrel.lancs.ac.uk, 2011) (Bandera, 2006) (git, 2022)

Referencias

(2022). Repositorio de github para el proyecto de teoria de la probabilidad. `https://github.com/jerom-1/proyecto-probabilidad`.

3Blue1Brown (2022). Solving wordle using information theory. `https://www.youtube.com/watch?v=v68zYyaEmEA&t=840s`.

Bandera, J. S. R. (9 de mayo de 2006). Modelo matemático para la tipificación en la clasificación de los alelos hla. *Departamento de Ingeniería Industrial y Mecánica. Escuela de Ingeniería y Ciencias, Universidad de las Américas Puebla*.

ucrel.lancs.ac.uk (2011). Frequency lists, chapter 2: Spoken and written english, list 2.1: Alphabetical frequency list: speech v. writing (lemmatized). `https://ucrel.lancs.ac.uk/bncfreq/flists.html`.