Complejidad Estocástica

F. Tusell*

Resumen

Se presenta una introducción simple y no matemática al trabajo que en el campo de la Teoría de la Probabilidad y Estadística o sus aledaños gira sobre la noción de complejidad. Se enfatiza el entronque de ideas como las propuestas por Kolmogorov y Rissanen con nociones de muy antiguo enraizadas en la filosofía occidental.

Palabras clave: Complejidad; inferencia estadística; teoría de la información; descripción de longitud mínima.

1 Una vieja idea. . .

Si Dios Nuestro Señor me hubiera consultado antes de crear el mundo, le hubiera recomendado que hiciera algo más sencillo.

(atribuido a Alfonso X el Sabio. Citado en [7])

Hemos de remontarnos al menos setecientos años para encontrar los precedentes de una idea que hoy es lugar común. William de Ockham (1290?–1349?) propuso como criterio para seleccionar lo que hoy llamaríamos modelos el prescindir de complicaciones innecesarias; el "no multiplicar las entidades sin necesidad." Entre dos posibles explicaciones de un mismo fenómeno, Ockham sugería así que retuviéramos la más simple. Un principio que se ha popularizado como "la navaja de Ockham."

Es difícil —tal vez imposible— justificar tal recomendación si pretendemos hacerlo con rigor. Se puede ver como una regla de economía intelectual. Pero ha de ser la adecuación entre modelo[1] y realidad lo que guíe nuestro esfuerzo, si somos realistas; no nuestra comodidad intelectual. ¿Por qué hemos de preferir explicaciones simples si el mundo real, en muchas de sus manifestaciones, parece extremadamente complejo (véase la cita que abre esta sección)?

*Departamento de Economía Aplicada III (Estadística y Econometría). Facultad de CC.EE. y Empresariales, Universidad del País Vasco, Avda. del Lehendakari Aguirre, 83, 48015 BILBAO. E-mail: ft@alcib.bs.ehu.es.

[1]Siendo acaso muy impreciso con el lenguaje, utilizo "modelo" para designar un mecanismo formalizable en ecuaciones matemáticas que suponemos "explica" un fenómeno.

Quizá la mejor línea de defensa argumental de la recomendación de Ockham pueda basarse en su extraordinario éxito. La búsqueda de explicaciones "simples" ha sido un criterio que ha guiado la perspicacia de los científicos casi invariablemente hacia "buenos" modelos: modelos con relativa gran capacidad explicativa que frecuentemente se funden armoniosamente con otros en unificaciones progresivamente mejores. Esto ha sucedido en Física y también en otras disciplinas.

2 . . .con un cabo suelto

Pero ¿qué es simple? Porque para seguir el consejo de Ockham necesitamos saber cuando uno de dos modelos es más simple que otro.

Hay casos en los que hay poca duda. Entre dos modelos que proporcionen predicciones igualmente buenas, si uno hace uso de todos los supuestos de otro y alguno adicional, preferiremos el primero. Hablaremos en tal caso de modelos anidados.

Pero esto es la excepción y no la regla. Más bien se nos presenta con frecuencia el caso de modelos "solapados" o incluso aparentemente "disjuntos." Se hace mucho más difícil en este caso decidir cuál es el más simple. Y el problema sólo puede complicarse cuando tenemos modelos estadísticos que ofrecen un grado diferente de explicación o ajuste de la evidencia empírica. ¿Qué debemos preferir: un modelo muy simple, que sólo imprecisamente parece dar cuenta del fenómeno de interés, u otro que logra gran precisión al coste de una complejidad mucho mayor?

¿Qué precio debemos pagar por la simplicidad en términos de adecuación de los resultados proporcionados por nuestro modelo a los datos reales? O, alternativamente, ¿qué complejidad adicional está justificada por un mejor ajuste a la evidencia?

Ejemplo 2.1 Consideremos el caso en que tratamos de establecer un modelo estocástico relacionando la talla y el peso de un colectivo de personas. Imaginemos N pares de valores $(\text{Talla}_i, \text{Peso}_i)$. Cabría imaginar relación lineal entre ambos, o una relación polinómica (que, a la luz de la naturaleza de los datos, presupondríamos fácilmente cúbica). Es decir, podemos pensar en las siguientes dos relaciones entre Talla y Peso:

$$\text{Peso}_i = \beta_0 + \beta_1 \text{Talla}_i + \epsilon \quad (1)$$

$$\text{Peso}_i = \beta_0 + \beta_1 \text{Talla}_i + \beta_2 (\text{Talla}_i)^2 + \beta_3 (\text{Talla}_i)^3 + \epsilon \quad (2)$$

Los β_i son parámetros y ϵ es una perturbación aleatoria inobservable que diluye la relación entre las dos magnitudes objeto de estudio: dos personas de la misma talla no necesariamente tienen el mismo peso.

Dos relaciones como (1) y (2) son un buen ejemplo de lo que hemos llamado dos modelos anidados. Si en (2) hacemos $\beta_2 = \beta_3 = 0$, tenemos una relación equivalente a (1).

No sólo podríamos pensar en dos relaciones como las citadas (la segunda de las cuales siempre proporcionará un mejor ajuste que la primera, si nos dejan escoger los parámetros). Podríamos pensar en una relación funcional ajustando *perfectamente* los datos. Por ejemplo, un polinomio de grado $N - 1$ (suponemos que no hay abscisas repetidas). Intuitivamente, parece que tal relación funcional es mucho más compleja, y aunque el ajuste *a los N puntos muestrales* fuera perfecto, seríamos bastante reticentes a aceptar un polinomio de grado muy elevado como modelo adecuado de una relación subyacente entre talla y peso.

El ejemplo anterior sugiere que el número de parámetros de un modelo es un candidato a medir su complejidad. También que, a mayor numero de parámetros —si trabajamos con modelos anidados—, mejor ajuste del modelo a los datos muestrales. Sin embargo, en una situación como la anterior podríamos acaso preferir una relación cúbica a una lineal —la mejora de ajuste quizá "vale" los dos parámetros adicionales de "complejidad"—, pero seríamos reticentes a admitir como modelo un polinomio de grado $N - 1$.

Este tipo de planteamiento se ha hecho desde largo tiempo, y hay un sin número de criterios de bondad de ajuste que dan orientaciones para dirimir el conflicto ajuste–simplicidad. Volveremos sobre ellos más tarde tras considerar brevemente las ideas de Kolmogorov, Chaitin y Solomonoff. A la luz de su contribución —y a la de la precedente y fundamental de Shannon— se puede ver el trabajo estadístico desde una nueva óptica, que ha encontrado un enérgico y brillante valedor en Rissanen (véase [14]).

3 La lógica máximo-verosímil

Previamente a abordar la noción de complejidad estocástica y su uso como criterio director en la selección de modelos, debemos hacer una breve incursión para describir la estimación máximo-verosímil. Lo haremos a través de un ejemplo.

> **Ejemplo 3.1** Consideremos dos urnas, urna I y urna II. La primera contiene 99 bolas blancas y 1 negra, en tanto la segunda contiene 50 bolas blancas y 1 negra. Nos ofrecen una de ellas, sin decirnos cuál es ni permitirnos observar su interior. Nos permiten sin embargo extraer una bola, que resulta ser negra, y nos preguntan ante que urna estamos.
>
> Diríamos, sin duda, que ante la urna II. La bola negra puede proceder de la urna I, pero ello sólo acontecería una vez de cada cien. ¿Por qué imaginar que ha ocurrido un suceso relativamente raro cuando, bajo el supuesto de que la urna es la II, lo observado (bola negra) acontece con relativa facilidad? Esta conclusión puede verse como una aplicación del criterio de máxima verosimilitud: *entre varios posibles "estados de la Naturaleza" (en el ejemplo, las dos urnas) suponer vigente aquél con óptima capacidad generadora de la evidencia o muestra con que contamos.*

Es interesante ver el parentesco del principio de máxima verosimilitud con la "navaja de Ockham." No es la misma cosa, pero sí muestra cierta similitud: evitar el pensar en sucesos infrecuentes cuando hay alternativas más plausibles que dan cuenta de lo que observamos es un modo de buscar simplicidad.

El Ejemplo 3.1 recoge la esencia de la lógica máximo-verosímil. En la práctica, sin embargo, los estados de la Naturaleza suelen estar rotulados mediante parámetros y los problemas de interés se traducen a problemas de estimación de parámetros o contraste de hipótesis sobre los mismos. De nuevo un ejemplo ilustra esto con simplicidad.

> **Ejemplo 3.2** Supongamos cien monedas, aparentemente idénticas, cada una de ellas con dos caras que denotamos por "cara" (C) y "cruz" (+). Imaginemos que cada una de ellas tiene

probabilidad θ de proporcionar C en un lanzamiento[2] y correlativa probabilidad $1-\theta$ de proporcionar '+'.

Lanzamos las cien monedas y obtenemos el resultado $\vec{x} = (x_1, \ldots, x_{100})$ con sesenta 'C' y cuarenta '+'. La Teoría de la Probabilidad indica que si la probabilidad de 'C' es θ, la probabilidad del suceso considerado[3] viene dada por,

$$P(\vec{x}|\theta) \quad = \quad \theta^{60}(1-\theta)^{40}, \tag{3}$$

y un sencillo cálculo muestra que el valor de θ que hace mayor (3) es $\theta = \frac{6}{10}$; el correspondiente valor de $P(\vec{x}|\theta)$ es $\approx 5.9085 \times 10^{-30}$. Llamamos verosimilitud de la muestra $\vec{x} = (x_1, \ldots, x_{100})$ a la expresión (3) vista como función de θ. El maximizar dicha expresión respecto de θ supone entonces escoger el valor del parámetro (estado de la Naturaleza) que hace más probable un suceso como el observado.

Una alternativa sería imaginar que cada moneda, pese a ser aparentemente idéntica a las restantes, tiene su propia probabilidad de proporcionar 'C' ó '+'. La expresión (3) se transformaría entonces en

$$P(\vec{x}|\theta) \quad = \quad \prod_i \theta_i \prod_j (1-\theta_j), \tag{4}$$

en que el primer producto consta de sesenta términos y el segundo de cuarenta. Siendo $0 \leq \theta \leq 1$, (4) se maximiza dando a θ_k, $k = 1, \ldots, 100$, valor 1 ó 0, según la moneda correspondiente haya proporcionado cara o cruz. El valor máximo de (4) es así 1.

Es poco natural atribuir a cada moneda una probabilidad θ_i de "cara" diferente, habida cuenta de que parecen iguales. Obviamente, al hacerlo maximizamos la probabilidad de observar algo como lo acontecido: ¡con la elección referida de los cien parámetros $\theta_1, \ldots, \theta_{100}$ el suceso observado pasaría a tener probabilidad 1, lo que hace el suceso casi seguro! Sin embargo, aparte de poco atractivo intuitivamente, el modelo es claramente más complejo que el que usa sólo un parámetro, y difícilmente sería adoptado por nadie. Y ello a pesar de que tendría óptima capacidad generadora de un resultado como el observado.

Como conclusión provisional de lo anterior, el criterio máximo verosímil es intuitivamente atrayente, aparte de tener propiedades muy deseables en grandes muestras (véase cualquier texto de Estadística, por ejemplo [10, 5]); pero no puede tomarse en consideración para comparar modelos cuya complejidad —en un sentido aún por determinar, pero que parece tener mucho que ver con el número de parámetros— es muy disimilar.

4 Teoría de la información

Precisamos de un último ingrediente antes de introducir la noción de complejidad según Kolmogorov-Chaitin-Solomonoff, y su aplicación, entre otras, estadística. Es la Teoría de la Información, para

[2] Con lo cual, para simplificar, queremos decir que imaginamos que en una sucesión muy larga de lanzamientos tenderíamos a observar un 100θ% de 'C' y el resto de '+'.

[3] Es decir, sesenta "caras" y cuarenta "cruces" *precisamente* en el orden en que han aparecido; si prescindiéramos de considerar el orden, la cifra dada habría de multiplicarse por $\binom{100}{60}$.

la que [17] (reimpreso en [18]) continúa siendo una referencia fundamental además de fácilmente accesible a no matemáticos. Otros textos introductorios son [1] y [6].

Supongamos una fuente aleatoria de símbolos $a_1, \ldots, a_k$ que genera una sucesión de los mismos con probabilidades respectivas $p_1, \ldots, p_k$. Supongamos que símbolos sucesivos se generan de modo independiente[4]. Nos planteamos el problema de codificar (por ejemplo, binariamente) el flujo de símbolos, de tal modo que la transmisión de los mismos pueda hacerse con el mínimo número de dígitos binarios en promedio.

La solución es bastante obvia, y no se separa de la que Samuel Morse adoptó sobre base intuitiva al diseñar el código que lleva su nombre: reservaremos palabras de código (dígitos binarios, o combinaciones de ellos) "cortas" a los símbolos que se presenten con gran probabilidad, y asignaremos las de mayor longitud a los símbolos más improbables. De este modo, gran parte del tiempo estaremos transmitiendo palabras de código cortas[5].

Shannon dio base matemática a esta intuición, obteniendo algunos resultados de gran interés. En lo que sigue, sólo se proporcionan versiones simplificadas de algunos de ellos, que no obstante retienen bastante de su interés y evitan complicaciones formales. Pero bastantes enunciados podrían ser más generales[6].

Central a la Teoría de la Información es el concepto de *entropía*. Si tenemos una fuente aleatoria como la aludida al comienzo de la sección, generando k símbolos independientemente unos de otros con probabilidades respectivas $(p_1, \ldots, p_k)$, la entropía de la fuente (o de la distribución asociada a ella) viene dada por

$$H(p) \stackrel{\text{def}}{=} -\sum_{i=1}^{k} p_i \log_2 p_i.$$

La función $H(p)$ tiene bastantes propiedades interesantes. Una de ellas, inmediata, es que se anula cuando la distribución de símbolos se hace causal —es decir, cuando un símbolo se genera con probabilidad 1 y el resto con probabilidad cero—. Alcanza su máximo cuando la distribución es lo más difusa posible —en el caso de una distribución discreta que puede dar lugar a k símbolos, cuando cada uno de ellos tiene probabilidad $\frac{1}{k}$ de aparecer—.

Un resultado muy fácil de demostrar[7] es el siguiente:

Teorema 4.1 *Para cualesquiera distribuciones discretas asignando respectivamente probabilidades* $(p_1, \ldots, p_k)$ *y* $(q_1, \ldots, q_k)$ *a* k *símbolos* $(a_1, \ldots, a_k)$*, se tiene:*

$$-\sum_{i=1}^{k} p_i \log_2 q_i \quad \geq \quad -\sum_{i=1}^{k} p_i \log_2 p_i. \tag{5}$$

[4]Es decir, que la fuente es de memoria nula. Se puede extender la teoría a fuentes markovianas en que este supuesto está ausente.

[5]Morse reservó el `.` para la letra `e`, muy frecuente en inglés, reservando para símbolos bastante más infrecuentes los códigos más largos (por ejemplo el cero, `0`, codificado mediante `---`).

[6]En particular, las distribuciones utilizadas podrían ser continuas en vez de discretas, y los logaritmos en cualquier base, en lugar de binarios.

[7]Véase por ejemplo [1], p. 30.

Tabla 1: Ejemplo de construcción de código de Fano-Shannon.

Símbolo	p_i	$P_i = \sum_{j<i} p_j$	P_i	$L(i) = \lceil -\log_2 p_i \rceil$	Código
a_1	0.500	0	0.**0**00000...	1	0
a_2	0.250	0.500	0.**10**0000...	2	10
a_3	0.125	0.750	0.**110**000...	3	110
a_4	0.125	0.875	0.**111**000...	3	111

Hay otros interesantes hechos en los que la entropía juega un papel central. Por ejemplo, la mejor codificación que podemos hacer de los símbolos $(a_1, \ldots, a_k)$ requiere en promedio un número de dígitos binarios por símbolo acotado inferiormente por $H(p)$. Esto es intuitivamente coherente con la interpretación ya aludida de la entropía: $H(p)$ muy baja, significaría distribución de las probabilidades de los símbolos muy concentrada (dando gran probabilidad a uno o unos pocos símbolos, y poca al resto). Ello permitiría codificar los pocos símbolos muy probables con palabras de código muy cortas, y sólo raramente hacer uso de palabras más largas (para los símbolos más improbables).

Ejemplo 4.1 *(código de Fano-Shannon)* Veamos un modo de hacerlo. Supongamos una fuente generando cuatro símbolos a_1, a_2, a_3, a_4 ordenados de acuerdo a sus probabilidades respectivas p_1, p_2, p_3, p_4. Supongamos que éstas son las que se recogen en la segunda columna del Cuadro 1. Sea $P_i = \sum_{j<i} p_i$ como se indica en el Cuadro 1. Las palabras de código se asignan tomando una parte de la expresión binaria de P_i de longitud $L(i)$ igual a $-\log_2 p_i$ redondeado a la unidad superior. Intuitivamente, es fácil ver que el código anterior es razonable: asigna palabras cortas a los símbolos más probables —que ocupan las primeras posiciones en la tabla— y progresivamente más largas al resto.

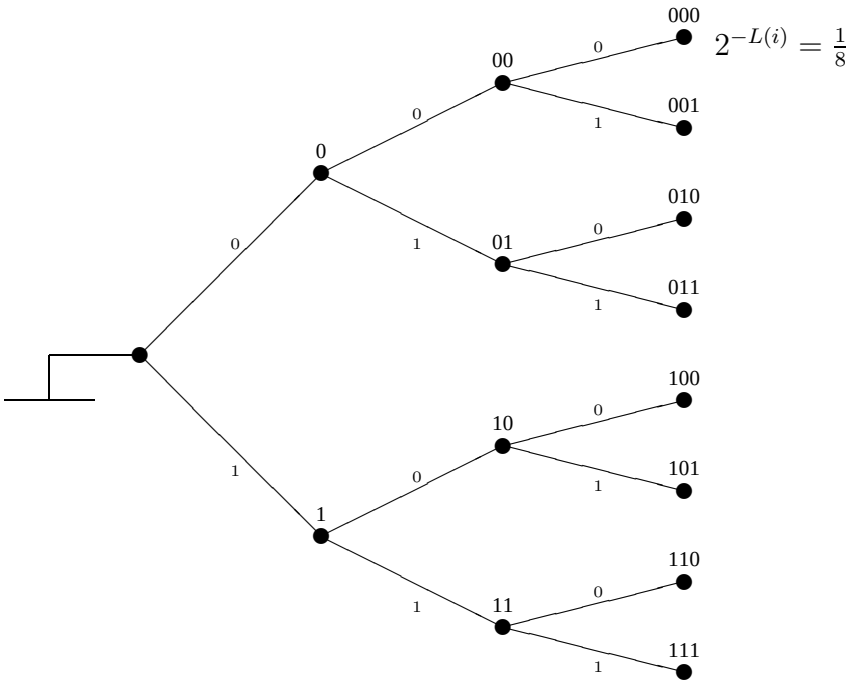
El código de Fano-Shannon comparte con otros una propiedad que se deriva fácilmente del proceso constructivo que hemos seguido (véase por ejemplo [11], p. 63) y que es aparente en la última columna del Cuadro 1: ninguna palabra de código es prefijo de otra de longitud mayor. Por ejemplo, a_2 se codifica por `10` que no es comienzo de ninguna de las dos palabras de código de longitud tres (`110` y `111`). Esta propiedad —la de ser un código *libre de prefijos* o *instantáneo* permite decodificar "al vuelo". Cuando observamos `10`, sabemos que hemos llegado al final de una palabra, que podemos decodificar como a_2; esto no ocurriría si nuestro código incluyera palabras como `101`.

Los códigos libres de prefijos tienen longitudes de palabra $L(i)$ verificando la llamada *desigualdad de Kraft*, recogida en el siguiente

Teorema 4.2 *La condición necesaria y suficiente para que exista un código libre de prefijos con longitudes de palabra $L(1), \ldots, L(k)$ es que*

$$\sum_i 2^{-L(i)} \leq 1 \tag{6}$$

Figura 1: Arbol binario completo de profundidad tres



DEMOSTRACIÓN:

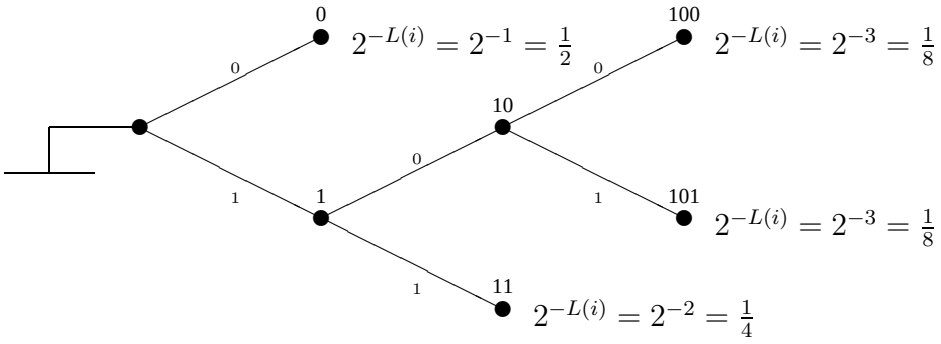
La demostración es muy simple. Pensemos en todas las posibles palabras de todas las longitudes dispuestas en un árbol binario como el recogido en el Gráfico 1 (truncado a la profundidad 3). Si utilizáramos como palabras de código todas las de longitud 3, tendríamos $L(i) = 3$ y $2^{-l(i)} = \frac{1}{8}$ para $i = 1, \ldots, 8$ y la inecuación (6) se verificaría con igualdad. Si escogemos una de las palabras de longitud inferior (uno de los nodos que no son "hojas" en el Gráfico 1), el requerimiento de ausencia de prefijos nos obliga a prescindir de todas las palabras correspondientes a nodos "hijos". El Gráfico 2 representa un árbol truncado con cuatro nodos terminales u "hojas", junto a las que se ha escrito $2^{-L(i)}$. Vemos que el tomar en `0` obliga a prescindir de `01`, `00`, y todos sus descendientes; pero 2^{-1} —contribución de `0` al lado izquierdo de (6)— es igual a la suma de las contribuciones a dicha expresión de todos los descendientes de los que hemos de prescindir.

Por tanto, trunquemos como trunquemos el árbol binario, la suma de $2^{-L(i)}$ extendida a sus "hojas" o nodos terminales será siempre 1. La desigualdad (6) sólo es estricta cuando despreciamos algún nodo terminal al construir nuestro código.

■

Podemos ya bosquejar la demostración del siguiente resultado:

Figura 2: Arbol binario truncado



Teorema 4.3 *Dada una fuente aleatoria con entropía* $H(p) = -\sum_i p_i \log_2 p_1$ *cualquier código instantáneo precisa un promedio de al menos* $H(p)$ *dígitos binarios de código por símbolo. Es decir, si la palabra codificando* a_i *tiene longitud* $L(i)$*, se verifica:*

$$\sum_i p_i L(i) \geq -\sum_i p_i \log_2 p_i \tag{7}$$

DEMOSTRACIÓN:
Definamos

$$q_i \;=\; \frac{2^{-L(i)}}{\sum_i 2^{-L(i)}}, \tag{8}$$

con lo que

$$\log_2 q_i \;=\; -L(i) - \log_2 \left(\sum_i 2^{-L(i)} \right) \geq -L(i). \tag{9}$$

La desigualdad anterior junto con el Teorema 4.1 proporcionan entonces de inmediato (7).

■

Obsérvese que el código de Fano-Shannon hacía $L(i) \approx -\log_2 p_i$ (redondeaba a la unidad superior): aproximadamente lo correcto. Verificaría (7) con igualdad si $-\log_2 p_i$ $(i = 1, \ldots, k)$ resultaran ser siempre números enteros. En cualquier caso, el resultado que nos interesa es que para codificar un evento de probabilidad p_i, el código libre de prefijos óptimo requiere del orden de $-\log_2 p_i$ dígitos binarios.

5 Complejidad en el sentido de Kolmogorov

5.1 Información y complejidad

Estamos ya en condiciones de abordar la noción de complejidad según Kolmogorov-Chaitin-Solomonoff.

De cuanto se ha visto en la Sección 4 se deduce que $\log_2 p_i$ mide aproximadamente la información contenida en a_i. Se da sin embargo una paradoja, ya puesta de manifiesto por Laplace (véase por ejemplo [4]), que sugiere emplear como medida de la complejidad de a_i algo diferente (aunque íntimamente relacionado con lo anterior).

Imaginemos las dos siguientes cadenas de dígitos binarios:

0000000000000000000000000000000

0011010001011101010001010111011

Ambas tienen el mismo número de dígitos binarios, 31. Si imaginamos el conjunto de todas las cadenas de 31 dígitos binarios —hay 2^{31} diferentes— y tomamos de ellas una al azar, cualquiera de las dos exhibidas tiene la misma probabilidad de aparecer: 2^{-31}. Sin embargo, desearíamos asignar a la primera una complejidad menor que a la segunda. Un modo de racionalizar esto es que podemos transmitir la primera a un tercero mediante una descripción muy parca: "treinta y un ceros." La segunda requiere una descripción más verbosa, que a duras penas podría ser más escueta que la cadena misma[8].

5.2 Complejidad de Kolmogorov

Una idea prometedora en línea con la discusión anterior fue propuesta en los años sesenta por Solomonoff, Kolmogorov y Chaitin, de manera independiente unos de otros y con ligeras variantes[9]. La *complejidad de Kolmogorov* de una cadena binaria x es la longitud del mínimo programa p capaz de computarla. Formalmente,

$$C_f(x) \;=\; \min\{l(p) : f(p) = x\}\,. \tag{10}$$

Por razones técnicas, f en (10) debe ser una función recursiva —el tipo de función que puede computar una máquina de Turing—. Naturalmente, el "programa" p que, al ser ejecutado por el "computador" f, produce la cadena x, depende de f. Sea cual fuere x, podríamos imaginar un "computador" especializado que tan pronto se pone en marcha imprime x —es decir, que requiere un programa de longitud $l(p) = 0$ para computar x. ¿Implicaría esto que la complejidad de x es cero?

No. La complejidad de x relativa a la máquina de Turing que computa f vendría dada por (10). Relativa a otra máquina de Turing computando la función g sería $C_g(x)$, definida análogamente a $C_f(x)$. Definiremos la complejidad de Kolmogorov en relación a una máquina de Turing universal —una máquina que con el programa adecuado puede emular cualquier otra—. No hay una única máquina universal, pero para dos máquinas universales de Turing computando las funciones u y v y para cualquier cadena x se verifica

$$|C_u(x) - C_v(x)| \leq c_{u,v}, \tag{11}$$

[8]Esto es lo que caracteriza a las cadenas binarias "típicas"; véase por ejemplo [11].

[9]La precedencia en el tiempo parece corresponder a Solomonoff: como en tantas otras ocasiones, la escena estaba preparadas en los años cincuenta para que investigadores trabajando de modo independiente llegarán a resultados similares. Véase una historia somera en [11], Sección 1.6.

en que $c_{u,v}$ es una constante que depende de u y de v, pero *no* de x.

> **Ejemplo 5.1** En [11] se propone una ilustración de lo anterior que ayuda a la intuición a ver el sentido de (11). Hay lenguajes de alto nivel especializados en cálculo numérico y en cálculo simbólico: `FORTRAN` y `LISP` serían dos buenos ejemplos. Cierto tipo de problemas pueden programarse muy fácilmente en `FORTRAN` y son considerablemente más farragosos en `LISP`; en otros ocurre lo contrario. Pero podríamos imaginar programar en `FORTRAN` un intérprete de `LISP` (requiriendo un programa de c_1 bits de longitud) y en `LISP` uno de `FORTRAN` (requiriendo a su vez una longitud de c_2 bits). Entonces, la diferencia de longitudes de programa para resolver un mismo problema en `FORTRAN` o `LISP` nunca excedería de $c_{F,L} = \max\{c_1, c_2\}$; $C_{F,L}$ sería el máximo "precio" a pagar para implementar el lenguage más favorable al problema a mano en el otro lenguaje. Este precio es independiente del programa que se desea ejecutar: una vez programado en `FORTRAN` un interprete de `LISP` podemos emplear éste para ejecutar programas en `LISP` de cualquier longitud.

Todas las máquinas de Turing universales (o, alternativamente, las funciones recursivas que computan) se agrupan en clases de equivalencia en que cada pareja de funciones verifica (11), para una constante que sólo depende de la pareja considerada. Se puede demostrar que existe una "clase mínima", en el sentido de que (11) no se verifica para ninguna constante $c_{u,v}$ si u pertenece a la clase mínima y v no. Entonces, $C_u(x)$ define (salvo una constante) la complejidad de una cadena binaria x.

5.3 $C_u(x)$ no es computable

El desarrollo anterior es útil por su poder clarificador, pero no directamente aplicable para computar un número que sea complejidad de una cierta cadena binaria. No existe un algoritmo con garantía de término que, al ser ejecutado por una máquina de Turing y alimentado con una cadena binaria, proporcione su complejidad.

No este el lugar para una discusión detallada de la no computabilidad de la complejidad de Kolmogorov, pero si puede intentarse una percepción intuitiva del motivo[10].

Imaginemos una cadena binaria x de n bits. Su complejidad no puede exceder mucho de n bits, ya que x es una descripción de sí misma. El programa más corto generando x no puede ser más largo que "print x", o su equivalente en la máquina de Turing de referencia que estemos empleando. Supongamos que la longitud de dicho programa es $(n + c)$ bits.

Podríamos ingenuamente pensar en formar una tabla con las cadenas binarias de longitud menor o igual que $(n+c)$, y ejecutarlas sucesivamente como programas en nuestra máquina de Turing, anotando si el resultado es x o no. Cada vez que obtuviéramos x, anotaríamos la longitud de la cadena binaria que hubiera servido como programa. Al final, la menor de las longitudes así anotadas, sería la complejidad de x.

Pero nada garantiza que haya final, porque nada garantiza que la máquina de Turing que empleamos se detenga al ejecutar como programa una cualquiera de las cadenas que le pasamos;

[10] Que sigue el razonamiento en el último capítulo de [15], una introducción muy legible y diáfana al tratar esta cuestión, aunque sólo lo haga tangencialmente al final.

mucho menos que lo haga con todas. La no computabilidad de $C_u(x)$ deriva del *halting problem*, o imposibilidad de determinar anticipadamente si una máquina de Turing se detendrá o proseguirá indefinidamente ejecutando un programa determinado. Sobre la no computabilidad de $C_u(x)$, y su relación con el teorema de Gödel y la indecidibilidad de proposiciones puede verse [11] y [3].

6 De la complejidad de Kolmogorov a la Longitud de Descripción Mínima (MDL)

Si bien no podemos hacer uso directamente de la complejidad de Kolmogorov para escoger entre distintos modelos, las ideas expuestas son de forma limitada aplicables. Veremos el modo de hacerlo sobre un ejemplo que, aunque artificialmente simple, ilustra la aproximación propuesta por Rissanen (véase [14]),

> **Ejemplo 6.1** *(continuación del 3.2)* Regresemos al Ejemplo 3.2. Describir llanamente el resultado de un experimento como el allí realizado al lanzar cien monedas al aire requiere 100 bits, si aceptamos el convenio de utilizar el dígito binario 0 para codificar el resultado '+' y el 1 para codificar el resultado 'C'. Obsérvese que 100 bits es exactamente la cantidad de información necesaria para singularizar una cadena binaria de longitud 100 de entre las 2^{100} posibles cuando no hay nada que haga unas de ellas más plausibles que otras.
>
> ¿Lo podemos hacer mejor? De acuerdo con lo visto en la Sección 4, podemos elaborar una codificación que atribuya a $\vec{x}$ una palabra de código de longitud $\lceil -\log_2 P(\vec{x}|\theta)\rceil$, si conocemos θ. Pero éste no es el caso. Para codificar $\vec{x}$ habremos de hacerlo con un valor de θ, y a continuación deberemos codificar este θ.

En lo que sigue formalizaremos algo esta idea.

6.1 Modelos como generadores de códigos

Consideremos una fuente aleatoria que ha generado $\vec{x}$. Si tenemos un modelo probabilístico, en general dependiente de parámetros $\vec{\theta}$, que describe el modo en que se genera $\vec{x}$, podemos calcular $P(\vec{x}|\vec{\theta})$ para los distintos resultados experimentales. Resultados con $P(\vec{x}|\vec{\theta})$ "grande" corresponderán a resultados esperables, que desearíamos claramente codificar mediante palabras de código cortas. Lo contrario ocurre con aquéllos en que $P(\vec{x}|\vec{\theta})$ es pequeño.

Estamos pensando como si θ fuera fijo y conocido, pero no lo es: lo hemos de escoger (estimar). Si lo hacemos maximizando $P(\vec{x}|\vec{\theta})$ (aplicando por tanto el principio de máxima verosimilitud), estamos atribuyendo al resultado $\vec{x}$ observado la máxima probabilidad. Pero no debemos olvidar que, para que sea posible la decodificación, hemos de facilitar también el valor $\vec{\theta}$ codificado (y la forma de nuestro modelo). El uso de máxima verosimilitud minimiza $\lceil -\log_2 P(\vec{x}|\vec{\theta})\rceil$, pero hace caso omiso de la longitud de código necesaria para $\vec{\theta}$.

6.2 Descripción de longitud mínima (MDL)

El agregar a $\lceil -\log_2 P(\vec{x}|\theta) \rceil$ el número de bits necesario para codificar los parámetros da lugar a la versión más cruda del llamado criterio MDL o de "mínima longitud de descripción."

A efectos de codificar los parámetros hemos de considerar dos cosas. En primer lugar, podemos tener información *a priori* sobre los mismos, de cualquier procedencia, traducible a una distribución a priori sobre los mismos con densidad $\pi(\vec{\theta})$. En segundo lugar, típicamente $\vec{\theta}$ es un número real que requeriría infinitos bits fijar con exactitud. Por ello trabajaremos con una versión truncada de él.

Si para el parámetro θ deseamos utilizar q dígitos binarios, llamaremos precisión a $\delta = 2^{-q}$. Suponiendo una densidad *a priori* $\pi(\theta)$, tendríamos los posibles valores de θ clasificados en intervalos de probabilidad aproximada $\pi(\theta)\delta$, especificar uno de los cuales requiere aproximadamente $-\log_2 \pi(\theta)\delta$ bits. Si hay k parámetros, se tiene la generalización inmediata,

$$-\log_2 \pi(\vec{\theta}) \prod_{i=1}^{k} \delta_i. \tag{12}$$

El criterio MDL propone tomar el modelo que minimiza la longitud total de código, la necesaria para los datos $\vec{x}$ más la necesaria para los parámetros:

$$\begin{aligned} MDL &= -\log_2 P(\vec{x}|\vec{\theta}) + l(\vec{\theta}) && (13) \\ &= -\log_2 P(\vec{x}|\vec{\theta}) - \log_2 \pi(\vec{\theta}) - \sum_{i=1}^{k} \log_2 \delta_i. && (14) \end{aligned}$$

en que $l(\theta)$ es la longitud de código necesaria para transmitir el o los parámetros empleados. Un ejemplo, de nuevo artificialmente simple, ilustra esto.

Ejemplo 6.2 *(continuación del Ejemplo 3.2)* Imaginemos que decidimos truncar el valor de θ en el Ejemplo 3.2 a 8 bits —por tanto sólo consideramos valores con una resolución de $\delta = 2^{-8} \approx 0.003906$—. Llamemos Θ_* al conjunto de valores que puede adoptar el parámetro así truncado. Imaginemos también que tenemos una distribución a priori uniforme $\pi(\theta)$ sobre los valores de θ; como $0 \leq \theta \leq 1$, $\pi(\theta) = 1$.

El criterio MDL para el modelo considerado en el Ejemplo 3.2 tomaría el valor:

$$MDL = \min_{\theta \in \Theta_*} \left\{ -\log_2 \theta^{60}(1-\theta)^{40} - \log_2 \pi(\theta) - \log_2 \delta \right\} \tag{15}$$

Si suponemos δ constante, sólo nos hemos de preocupar de minimizar el primer término. De poder escoger θ libremente, tomaríamos $\theta = 0.60$. Como estamos truncando los valores, 0.60 no es alcanzable, pero sí lo son $(153 + \frac{1}{2})/256 = 0.599609$ y $(154 + \frac{1}{2})/256 = 0.603516$, puntos medios de intervalos de longitud $1/256$ en que se subdivide $[0, 1]$ cuando se emplea precisión $\delta = 2^{-8} = 1/256$. El primero de ellos proporciona el mínimo valor de $-\log_2 P(\vec{x}|\theta)$, que resulta ser 97.0951. Requerimos un total de $97.0951 + 8 = 105.0951$ bits como longitud de descripción.

Una alternativa (tal y como se discutió a continuación del Ejemplo 3.2) sería considerar cien parámetros, uno para cada moneda. Ello haría "casi seguro" el suceso observado, y el

Tabla 2: Longitud de descripción para diferentes valores de δ.

q	δ	$\hat{\theta}_{MV}$	$\hat{\theta}$	$\hat{\theta}^{90}(1-\hat{\theta})^{10}$	$-\log_2 \hat{\theta}^{90}(1-\hat{\theta})^{10}$	MDL
1	0.50000	0.90	0.75	5.4314×10^{-18}	57.35	58.35
2	0.25000	0.90	0.875	5.6211×10^{-15}	47.34	49.34^*
3	0.12500	0.90	0.9375	2.7303×10^{-15}	48.38	51.38
4	0.06250	0.90	0.90625	7.447911×10^{-15}	46.93	50.93

primer sumando de (15) sería cero —especificados los parámetros, no haría falta ningún código para especificar el resultado—. Pero el tercer sumando sería, para la misma precisión, mucho mayor: ¡800 bits! Aunque el modelo binomial haciendo uso de cien parámetros hace casi seguro el resultado observado, es inferior al que sólo hace uso de sólo un parámetro, debido al coste de codificar noventa y nueve parámetros adicionales.

El ejemplo anterior suponía δ fijo a efectos puramente ilustrativos: pero en la práctica se minimiza MDL en (14) sobre θ y sobre δ. Es fácil ver que mientras disminuir la precisión (incrementar δ) disminuye el tercer sumando, hace en general crecer el primero (el "mejor" θ en Θ_* estará en general más lejos del óptimo θ cuanto más tosca sea la discretización de θ).

Un último ejemplo permitirá ver el efecto de optimizar la longitud de descripción sobre δ, precisión del parámetro.

Ejemplo 6.3 *(continuación de los Ejemplos 3.2, 6.1 y 6.2)* Consideremos la misma situación del Ejemplo 3.2, pero supongamos —para mostrar un caso en que se obtiene una reducción apreciable de la longitud de descripción— que se han obtenido noventa "caras" 'C' y diez '+'. Optimizaremos sobre $\delta = 2^{-q}$ dejando variar q sobre los enteros. El estimador máximo verosímil de θ es $\hat{\theta}_{MV} = 0.9$. El Cuadro 2 muestra el valor de θ entre los posibles que minimiza MDL para cada q. Con un asterisco se señala la descripción más escueta de los datos a que se llega. Obsérvese que cuando consideramos una precisión de $\delta = 2^{-q}$ estamos dividiendo $[0, 1]$ en 2^q intervalos del la forma $[n2^{-q}, (n+1)2^{-q})$ ($n = 0, 2^q - 1$), cuyo punto medio es $n2^{-q} + 2^{-q-1}$; estos son los valores que se recogen en la columna $\hat{\theta}$.

Obsérvese que aquí la longitud de descripción es acusadamente menor que los 100 bits que requeriría describir el resultado de nuestro experimento. Al ser uno de los resultados ('C') considerablemente más frecuente, podemos diseñar un código que tenga esto en consideración. No ocurría lo mismo en el Ejemplo 6.2, en que la ligera mayor probabilidad de 'C' dejaba poco margen a la optimización del código; como se vio, la ventaja obtenida no alcanzaba a "pagar" la especificación del parámetro necesario.

6.3 De la MDL a la complejidad estocástica

La discusión en el apartado anterior no hace sino introducir algunas ideas esenciales; pero en modo alguno hace justicia a la potencia del método.

La mínima longitud de descripción (MDL), en cierto sentido, es *más* de lo que buscábamos. Deseábamos una codificación compacta de $\vec{x}$ y hemos acabado con una codificación de $\vec{x}$ y *adicionalmente* de $\vec{\theta}$. La complejidad estocástica se obtiene integrando $P(\vec{x}|\vec{\theta})\pi(\vec{\theta})$ sobre los parámetros. En otras palabras, tenemos una distribución $P(\vec{x}|\vec{\theta})$ de los datos dados los parámetros y el modelo, y una densidad *a priori* $\pi(\vec{\theta})$ sobre los parámetros. La complejidad estocástica de los datos $\vec{x}$ relativa al modelo considerado se define como

$$I(\vec{x}) \;=\; \int_{\Theta} P(\vec{x}|\vec{\theta})\pi(\vec{\theta}) \tag{16}$$

(véase [14] para más detalles). Además, en el caso de que no tengamos una distribución *a priori* sobre los parámetros, podemos emplear la distribución *a priori* universal. Supongamos que deseamos una codificación que asigne una palabra de código a todos los números naturales n, sobre los que hay definida una distribución $P(n)$. Bajo condiciones muy generales, existe una codificación asignando longitud de palabra $L^*(n)$ a n y que verifica

$$\lim_{N\to\infty} \frac{\sum_{i=0}^{N} P(n)L^*(n)}{\sum_{i=0}^{N} P(n)\log_2 n} \;=\; 1 \tag{17}$$

Merece la pena examinar la igualdad anterior: ¡hay una codificación que es asintóticamente óptima sobre los enteros y que es "todo terreno"! ¡Vale sea cual fuere la distribución definida sobre ellos, con tal de que sea monótona decreciente a partir de algún n dado! La función $L^*(n)$ viene dada aproximadamente por

$$L^*(n) \;=\; \log_2 c + \log_2\log_2 n + \log_2\log_2\log_2 n + \ldots, \tag{18}$$

con $c = 2.865$ verifica la desigualdad de Kraft y a partir de ella puede obtenerse una distribución *a priori* universal: $P(n) = 2^{-L^*(i)}$. Esta es la que Rissanen propone utilizar en la definición de complejidad estocástica[11]. En el caso en que tenemos parámetros que no toman valores enteros, se puede también definir una distribución *a priori* universal del modo descrito en [13].

6.4 Ideas relacionadas y conexas

Aunque en el Ejemplo 6.3 se ha buscado la longitud de descripción minimizando explícitamente sobre la precisión (en el Cuadro 2), en la práctica no es preciso recorrer un camino similar con cada modelo que se prueba. Argumentos de tipo asintótico dan un resultado similar en forma mucho más simple. Habitualmente sólo se requiere computar una función que da aproximadamente la longitud de descripción, y que típicamente consta de una parte que disminuye al mejorar el ajuste a los datos (término de fidelidad o ajuste) y otra que crece con el número de parámetros (término de penalización de la complejidad del modelo). Por ejemplo, de modo bastante general (véase [14] para las condiciones necesarias) la mínima longitud de descripción de $\vec{x} = (x_1, \ldots, x_N)$ utilizando un modelo con p parámetros viene dada por:

$$\mathrm{MDL}(p) \;=\; -\log\left(P(\vec{x}|\hat{\theta})\pi(\hat{\theta})\right) + \frac{p}{2}\log N + O(p). \tag{19}$$

[11] En el Ejemplo 6.2 hemos empleado una densidad $\pi(\theta)$ uniforme por simplicidad.

Puede verse un primer término que disminuye al mejorar el ajuste y un segundo término (la penalización) que crece con el número de parámetros p y está dominado por $\frac{p}{2}\log N$.

A la vista de una expresión como (19) es forzoso pensar en los muchos criterios que se han propuesto para evaluar la adecuación de un modelo, muchas veces sobre bases puramente heurísticas. En el caso de modelos de regresión lineal tenemos por ejemplo el estadístico conocido como C_p de Mallows,

$$C_p \;=\; \frac{\sum_{i=1}^{N}\hat{\epsilon}^2}{\hat{\sigma}^2} + 2p \tag{20}$$

en que $\hat{\epsilon}$ son los residuos de la regresión y σ^2 la varianza del término de error: véase [12]. El primer término de (20) disminuye al mejorar el ajuste o fidelidad del modelo a los datos; el segundo, crece con el número de parámetros.

En el análisis de series temporales ha sido también habitual el uso del llamado criterio de información de Akaike (véase [2]) definido por

$$AIC(p) \;=\; -2\log_e(P(\vec{x}|\hat{\theta})) + 2p, \tag{21}$$

también de la misma forma que (19), aunque penalizando asintóticamente menos la introducción de parámetros. Los ejemplos podrían multiplicarse; una recopilación reciente de trabajos incorporando ideas como las mencionadas a múltiples campos es [8].

La búsqueda de longitudes de descripción mínimas o mínimas complejidades no se separa pues, por lo menos asintóticamente, de algunos criterios que han sido utilizados con asiduidad. La novedad está más bien en la justificación de resultados antes obtenidos para problemas concretos y de forma bastante *ad-hoc* desde una perspectiva unificadora.

7 ¿Tiene sentido esto?

Hemos recogido e ilustrado ideas que basan la elección de modelos en un criterio de simplificación de la información. Apoyándose en el trabajo pionero que sobre la noción de complejidad y sobre Teoría de la Información se realizó en los años cincuenta y sesenta, estas ideas pueden verse como una navaja de Ockham sofisticada, de posible utilización en el trabajo estadístico. Importa ahora no obstante regresar al origen y preguntarse sobre el alcance, pertinencia y solidez de este modo de actuar.

¿Es la noción de complejidad de Kolmogorov —o versiones menos ambiciosas de la misma idea, como la de Rissanen— el anclaje al que deseamos asirnos para hacer inferencia? No lo sé. Es un planteamiento no exento de belleza, y que, como se ha indicado, da en su aplicación práctica resultados satisfactorios.

¿Debemos entender por complejidad sólo esto, o algo más? ¿Es la longitud de descripción tal como la hemos presentado una buena medida de la complejidad de un modelo más los datos, haciendo abstracción —por ejemplo— del coste de llegar a obtenerlo? Murray Gell-Mann (véase [9], p. 117) menciona, haciéndose eco de trabajo de Charles Bennet, que la complejidad tiene facetas como la *profundidad* y *cripticidad*. En relación a esta última, por ejemplo, una serie muy larga de

números pseudo-aleatorios generados en un ordenador mediante el conocido método multiplicativo, puede tener una complejidad muy baja: se puede describir dando la semilla o valor inicial y los valores de tan sólo dos números. Sin embargo, adivinar cuáles son estos números es muy costoso. ¿Diríamos que esta serie es de baja complejidad?

Adicionalmente, tenemos el problema filosófico de la posibilidad de conocer sobre la única base de la experiencia, de un modo inductivo. Si repetidamente, en una muestra de entrenamiento que ha permitido construir un modelo, y luego en datos distintos a dicha muestra, obtenemos un buen ajuste, se incrementa nuestra confianza subjetiva de estar ante un "buen" modelo. Incrementamos nuestro conocimiento —o creemos hacerlo— por inducción. Es en este sentido en el que la observación de que cada día sale el sol nos hace incrementar nuestra convicción de que saldrá mañana.

Un modelo es un modo de especificar regularidades. Decimos que "explica" la realidad cuando lo que observamos se adecúa a las predicciones que obtendríamos con ayuda de dicho modelo. En el caso de un modelo estadístico, ni siquiera exigimos una concordancia perfecta entre predicciones y observaciones, porque la esencia de un modelo de tal naturaleza es no fijar unívocamente las relaciones entre observables.

Es precisamente la existencia de regularidad en la evidencia lo que permite su descripción escueta. Servirse de un criterio como el de mínima longitud de descripción es aceptar como buena la "explicación" que más regularidades encuentra en nuestros datos —o mejor las explota—. Tiene al menos la ventaja sobre la modelización usual de que explicita el coste a pagar por la complejidad añadida. Queda a medio camino entre la inferencia bayesiana y la convencional, y sortea algunos de los aspectos más criticables en esta última —la fijación arbitraria de niveles de significación, por ejemplo—.

Pero, en su raíz, el minimizar la complejidad es un criterio que prioriza la reducción de los datos observados. ¿Es esto sensato? ¿Válido como criterio de inferencia?

B. Russell (véase [16], p. 35) obliga a responder que no. Un pollo que observara al granjero llevarle grano todos los días —dice Russell—, podría llegar a la conclusión de que el granjero le ama y busca su bien. Tal "modelo" explicaría las repetidas visitas al corral del granjero y su solicitud con el animal. Pero esta "explicación", tan repetidamente apoyada por la evidencia durante la vida del pollo, se ve bruscamente sin valor el día que el granjero decide que el pollo está lo suficientemente gordo como para retorcerle el pescuezo.

Enfrentados al mundo, querríamos saber *porqué*, y ni tan solo sabemos si nuestra noción de causalidad tiene sentido; si cabe hablar de un porqué. Querríamos conocer el fin último, si lo hay, de las idas y venidas del granjero: conformarnos con la explicación menos compleja de su conducta nos coloca en situación no mejor que la del pollo.

Sin embargo, frecuentemente no podemos hacer más. Enfrentados a este hecho, nuestra pertinaz tentativa de entender encuentra en el criterio de minimizar la longitud de descripción un sucedáneo útil: la vieja navaja de Ockham, de noble pátina, con un nuevo filo. El éxito que alcancemos con su empleo no debiera hacernos olvidar lo endeble de nuestra posición. Quizá el mayor valor de las ideas expuestas más arriba no esté en las respuestas que proporcionan sino en las preguntas que suscitan.

Referencias

[1] N. Abramson. *Teoría de la Información y Codificación*. Paraninfo, Madrid, 1973 edition, 1966.

[2] H. Akaike. Use of an information theoretic quantity for statistical model identification. In *Proc. 5th. Hawai Int. Conf. on System Sciences*, pages 249–250, 1972.

[3] G.J. Chaitin. *Algorithmic Information Theory*. Cambridge University Press, Cambridge, 1992 edition, 1987.

[4] T.M. Cover, P. Gacs, and R.M. Gray. Kolmogorov's contributions to information theory and algorithmic complexity. *Annals of Probability*, 17(3):840–865, 1989.

[5] D. R. Cox and D. V. Hinkley. *Theoretical Statistics*. Chapman and Hall, London, 1979 edition, 1974.

[6] G. Cullman, M. Denis-Papin, and A. Kaufmann. *Elementos de Cálculo Informacional*. Ed. Urmo, Bilbao, 1967 edition, 1967.

[7] Manuel do Carmo Gomes. *Predictions under Uncertainty. Fish Assemblages and Food Webs on the Grand Banks of Newfoundland*. ISER, Memorial University of Newfoundland, St. John's, Nfld., 1993.

[8] D.L. Dowe, K.B. Korb, and J.J. Oliver, editors. *Information, Statistics and Induction in Science – ISIS'96*, Melbourne, Australia, August 1996. World Scientific, Singapore.

[9] M. Gell-Mann. *El quark y el jaguar*. Tusquets, Barcelona, 1995 edition, 1994.

[10] E. L. Lehmann. *Theory of Point Estimation*. Wiley, New York, 1983.

[11] Ming Li and P. Vitányi. *An introduction to Kolmogorov complexity and its applications*. Springer-Verlag, New York, 1993.

[12] C.L. Mallows. Some comments on C_p. *Technometrics*, 15:661–675, 1973.

[13] J. Rissanen. A universal prior for integers and estimation by minimum description length. *Annals of Statistics*, 11(2):416–431, 1983.

[14] J. Rissanen. *Stochastic Complexity in Statistical Inquiry*. World Scientific, Singapore, 1989.

[15] D. Ruelle. *Chance and Chaos*. Penguin, London, 1991.

[16] B. Russell. *The problems of philosophy*. Oxford University Press, 1989 edition, 1912.

[17] C.E. Shannon. The mathematical theory of communication. *Bell System Tech. Journal*, 27:379–423, 623–656, 1948.

[18] C.E. Shannon and W. Weaver. *The mathematical theory of communication*. University of Illinois Press, Urbana, 1949. Eight reprint, 1980.