Sección didáctica: Principio *predictibilidad–novedad* en sistemas caóticos

Alberto

1. Por qué merece la pena formularlo

Creemos que esta idea puede aportar a la comunidad: en caos se dice que pequeñas diferencias iniciales crecen rápido y limitan la predicción. Pero en clase queda difuso *cuánto* podemos predecir y *cuánta novedad* (información nueva) se crea en cada paso. Por eso queremos **formular matemáticamente** un principio que relacione *predictibilidad* con *novedad*.

Para explicarlo usaremos nuestro **juguete favorito** en el estudio del caos: la **función logística**

$$x_{n+1} = r\,x_n(1-x_n), \qquad \text{y trabajaremos en el caso } r = 4,$$

que es caótico en todo $[0,1]$ y tiene propiedades exactas conocidas (es un caso de libro).[1]

2. Tiempo de predictibilidad: lo que ya sabíamos

Si el error inicial es ε (precisión con que conocemos el estado) y dejamos de considerar útil la predicción cuando el error supera Δ (umbral de utilidad), el crecimiento exponencial del error lleva a la fórmula operativa

$$\boxed{T_p(\varepsilon,\Delta) = \frac{1}{\lambda}\ln\frac{\Delta}{\varepsilon}}$$

donde λ es el **exponente de Lyapunov**. El inverso $1/\lambda$ se conoce como *tiempo de Lyapunov* (marca la escala temporal de la pérdida de predictibilidad).[2]

3. Qué nos falta: medir la "novedad"

Lanzamos la pregunta a ChatGPT: £cómo medir matemáticamente la creación de novedad? La sugerencia fue usar la **entropía por paso** (entropía de Kolmogórov–Sinai, h_{KS}), que cuantifica cuánta información nueva genera el sistema en cada iteración. En muchos sistemas bien comportados (medida natural), h_{KS} coincide con la suma de los exponentes de Lyapunov positivos (fórmula de Pesin); en mapas unidimensionales expansivos como los que veremos, vale $h_{\mathrm{KS}} = \lambda$.

En el **caso logístico** $r = 4$ ocurre algo especialmente limpio: está (semi)conjugado al *mapa tent* de pendiente 2 y al *bit-shift*/doblado $x \mapsto 2x$ mód 1. Eso significa que cada iteración libera exactamente 1 bit de novedad, y por tanto

$$\boxed{h_{\mathrm{KS}} = \lambda = \ln 2 \ (\text{nats/iter})}.$$

[1] Entrada *Logistic map*, secc. "When $r = 4$" y "Topological conjugate mapping". https://en.wikipedia.org/wiki/Logistic_map

[2] Sobre Lyapunov y predictibilidad, v. gr., *ChaosBook*, capítulo de Lyapunov. https://chaosbook.org/chapters/Lyapunov.pdf

Así, en la logística ($r = 4$) el **principio predictibilidad–novedad** se cumple al 100 %:

$$\boxed{h_{\mathrm{KS}}\, T_p(\varepsilon, \Delta) = \ln \frac{\Delta}{\varepsilon}}.$$

Lectura: más novedad por paso (h grande) implica menos tiempo útil de predicción (T_p pequeño), con una constante que solo depende de ε (precisión inicial) y Δ (tolerancia).[3]

4. Otros sistemas donde *ChatGPT* enumeró fórmulas explícitas

Le pedimos a ChatGPT más sistemas de libro donde haya fórmulas claras de entropía y, por tanto, del tiempo de predictibilidad. La **enumeración** que nos dio fue:

- **Doblado/Bernoulli shift** $T(x) = 2x \text{ mód } 1$
  Entropía y Lyapunov: $h_{\mathrm{KS}} = \lambda = \ln 2$. Predictibilidad: $T_p = \dfrac{1}{\ln 2} \ln \dfrac{\Delta}{\varepsilon}$.[4]
- **Mapa tent (carpa)** de pendiente $s \in (1, 2]$
  Entropía/Lyapunov: $h_{\mathrm{KS}} = \lambda = \ln s$. Predictibilidad: $T_p = \dfrac{1}{\ln s} \ln \dfrac{\Delta}{\varepsilon}$.[5]
- **Mapa del panadero (Baker)** (versión binaria; estiramiento b)
  Entropía KS: $h_{\mathrm{KS}} = \ln 2$ (o, en general, $\ln b$). Predictibilidad: $T_p = \dfrac{1}{\ln 2} \ln \dfrac{\Delta}{\varepsilon}$ (o $\dfrac{1}{\ln b} \ln \dfrac{\Delta}{\varepsilon}$).[6]
- **Gato de Arnold** (automorfismo toral hiperbólico)
  Entropía KS: $h_{\mathrm{KS}} = \ln \sigma_+$, con σ_+ el autovalor inestable de la matriz. Predictibilidad: $T_p = \dfrac{1}{\ln \sigma_+} \ln \dfrac{\Delta}{\varepsilon}$.[7]

5. Qué pasa con la información en un sistema caótico

En un sistema caótico partimos de un **estado inicial**; tras avanzar, tenemos un **estado final + novedad**. A diferencia de un sistema no caótico, no podemos reconstruir el pasado a partir del presente: en mapas no invertibles (como la logística) varios pasados llevan al mismo presente; en sistemas disipativos, volúmenes se "aplastan" hacia atractores; y con resolución finita la información observable del pasado se diluye por el estirar–plegar.

6. Cómo escribirlo de forma sencilla: el "presupuesto"

Preguntamos a ChatGPT cómo poner esto en una fórmula simple. La propuesta (didáctica) es el siguiente **balance** a resolución finita:

$$\underbrace{I_\varepsilon(X_0; X_t)}_{\text{información del pasado que aún queda}} + \underbrace{h_\mu\, t}_{\text{novedad acumulada (nats)}} \approx \underbrace{H_\varepsilon(X_0)}_{\text{información inicial a resolución } \varepsilon}.$$

[3] Conjugación logística $r = 4$ con tent ($s = 2$) y con el doblado/bit-shift; y $\lambda = h_{\mathrm{KS}} = \ln 2$. Apuntes introductorios: Phys 221A (UCSD). https://courses.physics.ucsd.edu/2017/Spring/physics221a/Phys_221A_Lecture_4-5.pdf

[4] Notas de curso con cálculo de entropía del doblado; ejemplo canónico. https://people.maths.bris.ac.uk/~ip13935/dyn/CorinnaII.pdf

[5] Derivación elemental: $|T'| = s$ casi en todas partes $\Rightarrow \lambda = \int \ln |T'| = \ln s$. Apuntes: Phys 221A (UCSD). https://courses.physics.ucsd.edu/2017/Spring/physics221a/Phys_221A_Lecture_4-5.pdf

[6] Apuntes con el cálculo directo de h_{KS} para Baker. https://courses.physics.ucsd.edu/2017/Winter/physics200b/LECTURES/CH02_MAPS.pdf

[7] Ficha y notas sobre el cat map; para la matriz estándar $\sigma_+ = (3 + \sqrt{5})/2$. https://mathworld.wolfram.com/ArnoldsCatMap.html

Aquí $H_\varepsilon(X)$ es la entropía de Shannon del estado *cuantizado* a celdas de tamaño ε; $I_\varepsilon(X_0;X_t)$ es la *información mutua* entre pasado y presente a esa misma resolución; y h_μ es la *entropía por paso* (novedad por iteración).[8]

Cuando I_ε cae a cero, se agota la predictibilidad y recuperamos el horizonte:

$$h_\mu\, T_p(\varepsilon,\Delta)\ \approx\ \ln\frac{\Delta}{\varepsilon}.$$

7. Prueba escolar perfecta en la logística $r=4$: se cumple *exacto*

Usamos cuantización binaria ($\varepsilon=2^{-k}$): conocer X_0 a esa resolución equivale a conocer los *primeros k bits*. Como la logística $r=4$ es (semi)conjugada al *bit-shift* (y al tent con pendiente 2), cada iteración *expulsa* el bit más significativo del pasado e *introduce* un bit nuevo (novedad). Por tanto:

$$\boxed{H_\varepsilon(X_0)=k\ln 2},\qquad \boxed{I_\varepsilon(X_0;X_t)=\max\{k-t,0\}\,\ln 2},\qquad \boxed{h_\mu\, t=t\ln 2}.$$

Suma paso a paso (para $0\le t\le k$):

$$I_\varepsilon(X_0;X_t)+h_\mu t=(k-t)\ln 2+t\ln 2=\boxed{k\ln 2}=\boxed{H_\varepsilon(X_0)}.$$

Exacto. Para $t\ge k$, $I_\varepsilon=0$: hemos agotado los k bits iniciales. Si fijamos un umbral $\Delta=2^{-m}$ ("me vale mientras conserve m bits"), el horizonte es

$$T_p(\varepsilon,\Delta)=k-m,\qquad h_\mu T_p=\ln 2\,(k-m)=\ln\frac{2^{-m}}{2^{-k}}=\ln\frac{\Delta}{\varepsilon}.$$

(En tent/doblado $h_{\mathrm{KS}}=\lambda=\ln 2$ y la dinámica es literalmente un desplazamiento de bits.)[9]

Ejemplo extendido: el termómetro y el horizonte de predictibilidad

Planteamiento base

Medimos temperatura con un termómetro de resolución inicial ε (por ejemplo, 0,1 °C). Consideramos que la predicción deja de ser útil cuando el error supera un umbral Δ (por ejemplo, 1,0 °C). Si el error crece aproximadamente de forma exponencial, $\varepsilon(t)=\varepsilon\, e^{\lambda t}$, el **tiempo de predictibilidad** cumple

$$T_p(\varepsilon,\Delta)=\frac{1}{\lambda}\,\ln\frac{\Delta}{\varepsilon}.$$

Aquí λ es el *exponente de Lyapunov* (tasa a la que divergencias pequeñas se amplifican); su inverso $T_\lambda=1/\lambda$ es el *tiempo de Lyapunov* o *e-folding time*.[10]

[8] Definiciones intro: entropía e información mutua (Wikipedia); romper URLs: paquete `xurl`. https://en.wikipedia.org/wiki/Entropy_(information_theory) https://en.wikipedia.org/wiki/Mutual_information https://tex.stackexchange.com/questions/3033/forcing-linebreaks-in-url

[9] Notas Phys 221A sobre tent y $2x$ mód 1 con $h_{\mathrm{KS}}=\ln 2$. https://courses.physics.ucsd.edu/2017/Spring/physics221a/Phys_221A_Lecture_4-5.pdf

[10] Definiciones y uso: Wikipedia, "Lyapunov exponent" y "Lyapunov time"; ChaosBook (cap. Lyapunov). En esas fuentes se explica que T_λ es el tiempo en el que el error se multiplica por e, y que también se usan doblajes (factor 2) o "décuplos"(factor 10) como escalas informativas. https://en.wikipedia.org/wiki/Lyapunov_exponent https://en.wikipedia.org/wiki/Lyapunov_time https://chaosbook.org/chapters/Lyapunov.pdf

Caso A (base): $\varepsilon = 0{,}1°\text{C}$**,** $\Delta = 1{,}0°\text{C}$**,** $T_p = 7$ **días**

Si observamos empíricamente un horizonte $T_p = 7$ días con esos umbrales, entonces

$$\lambda = \frac{1}{7}\ln\frac{\Delta}{\varepsilon} = \frac{\ln 10}{7} \approx 0{,}329 \text{ día}^{-1}, \qquad \text{factor diario } e^{\lambda} \approx \exp\Big(\tfrac{\ln 10}{7}\Big) \approx 1{,}3895.$$

Interpretación: cada día el error se multiplica por 1,3895; en 7 días se multiplica por 10 y pasa de 0,1 a 1,0 °C.

Tabla día a día (crecimiento del error y "consumo" de información) Sea $\varepsilon_d = \varepsilon_0\, e^{\lambda d}$ con $\varepsilon_0 = 0{,}1\,°\text{C}$. La información útil a la resolución del termómetro puede verse, de forma didáctica, como un presupuesto que se agota linealmente hasta T_p (en nats): $I_d \approx \ln(\Delta/\varepsilon)\,(1 - d/7)$.

Día d	Error ε_d (°C)	I_d (nats)	% de la info inicial
0	0,1000	2,3026	100,0 %
1	0,1389	1,9736	85,7 %
2	0,1931	1,6447	71,4 %
3	0,2683	1,3158	57,1 %
4	0,3728	0,9868	42,9 %
5	0,5179	0,6579	28,6 %
6	0,7197	0,3289	14,3 %
7	**1,0000**	**0,0000**	**0,0 %**

Reglas de oro (muy útiles para diseñar sensores/umbrales) Con λ fijo, el horizonte cambia *logarítmicamente* con los umbrales:

$$\Delta \text{ en factor } f \;\Rightarrow\; T_p \text{ varía en } \frac{1}{\lambda}\ln f, \qquad \varepsilon \text{ en factor } f \;\Rightarrow\; T_p \text{ varía en } -\frac{1}{\lambda}\ln f.$$

Ejemplos numéricos con $\lambda = 0{,}329\ \text{d}^{-1}$:

- Mejorar la resolución inicial de 0,1 a 0,05 ($f = 2$) $\Rightarrow$ *ganar*$\ln 2_{\overline{0{,}329 \approx 2{,}10}}$ días.
- Endurecer el umbral de utilidad de 1,0 a 0,5 ($f = 1/2$) $\Rightarrow$ *perder*$\ln 2_{\overline{0{,}329 \approx 2{,}10}}$ días.

Caso A: $\Delta = 10$ °C y $T_p = 14$ días

Derivación breve. Del balance didáctico $I_\varepsilon + h_\mu t \approx H_\varepsilon$ y usando $h_\mu \approx \frac{1}{T_p}\ln(\Delta/\varepsilon)$ y $H_\varepsilon \approx \ln(\Delta/\varepsilon)$, obtenemos a tiempo $t = d$ (en días):

$$I_d \;\approx\; \ln\frac{\Delta}{\varepsilon}\Big(1 - \frac{d}{T_p}\Big).$$

Con $\Delta = 10$ y, para fijar ideas, $\varepsilon = 1$, resulta $H_\varepsilon = \ln 10 = 2{,}3026$ nats y $h_\mu = \frac{\ln 10}{T_p} = \frac{2{,}3026}{14} \approx 0{,}16447$ nats/día.

Tabla (día a día). Columnas: *Información del pasado que queda* (I_d), *Novedad generada* ($h_\mu d$) y *Información total* ($I_d + h_\mu d = H_\varepsilon$). Valores en nats, redondeo a 4 decimales.

Día d	I_d	Novedad generada $h_\mu d$	Información total H_ε
0	2.3026	0.0000	2.3026
1	2.1381	0.1645	2.3026
2	1.9736	0.3289	2.3026
3	1.8092	0.4934	2.3026
4	1.6447	0.6579	2.3026
5	1.4802	0.8224	2.3026
6	1.3158	0.9868	2.3026
7	1.1513	1.1513	2.3026
8	0.9868	1.3158	2.3026
9	0.8224	1.4802	2.3026
10	0.6579	1.6447	2.3026
11	0.4934	1.8092	2.3026
12	0.3289	1.9736	2.3026
13	0.1645	2.1381	2.3026
14	0.0000	2.3026	2.3026

Lectura: la información del pasado I_d desciende linealmente desde $\ln 10$ hasta 0 en 14 días, mientras que la *novedad acumulada* crece al ritmo constante $h_\mu = \ln(10)/14$ nats/día. La suma se mantiene constante e igual a $H_\varepsilon = \ln 10$.

Notas técnicas útiles (LaTeX)

Para evitar "Missing character" con comillas , usa comillas LaTeX: "texto". Para URLs largas, el paquete `xurl` permite partir enlaces en cualquier punto; también ayuda `\sloppy`.[11]

[11] Comillas correctas en LaTeX: John D. Cook, "Top four LaTeX mistakes". https://www.johndcook.com/blog/2010/02/15/top-latex-mistakes/. Partir URLs con `xurl`: https://tex.stackexchange.com/questions/3033/forcing-linebreaks-in-url; guía práctica: https://www.baeldung.com/cs/latex-show-url.