Felipe Contreras
7 de febrero de 2020

Series de Tiempo

Para Complejidad 2 / SC2

Antecedentes

Recordando, las *Series de Tiempo* son colecciones de datos recolectados a intervalos regulares de tiempo. Por ejemplo, si tenemos los tiempos t_1, t_2, …, t_n, podemos tener los datos asociados a esos tiempos x_1, x_2, …, x_n. Es decir, tenemos una colección de datos, pero que están dados en cierto orden "cronológico", y un dato del siguiente están distanciados en el tiempo en una unidad fija (en nuestro ejemplo, $k = t_i - t_{i-1} = t_j - t_{j-1}$). Por tanto, basta saber el tiempo inicial t_1 y el intervalo k para determinar la componente temporal de toda la Serie de Tiempo (pues $t_{i+1}=t_1+ik$). Así, lo más importante de de la Serie de Tiempo son los datos precisamente capturados en esos intervalos de tiempo.

Se ha hablado que cualquier colección de datos es susceptible de ser tratada como Serie de Tiempo si se le toma de una manera "conveniente" y se le asigna tiempo inicial e intervalo. Así por ejemplo, la colección de datos consistente en las alturas en décimas de milímetro de las letras, símbolos y espacios de este texto podría ser tratada con técnicas de Series de Tiempo y así por ejemplo, hacer análisis sobre el texto aquí descrito.

Pero hablaremos en adelante sobre colecciones de datos con componente temporal dada. Un ejemplo es la colección de datos del Índice Nacional de Precios y Cotizaciones, que varía por segundo en días y horarios laborables.

Otro ejemplo son los datos provenientes de los movimientos voluntarios o no de los músculos del cuerpo de animales. Se sabe que a una señal del cerebro se puede desencadenar una serie de estos movimientos, por ejemplo, para levantar un brazo, se tiene una etapa inicial, una intermedia en la que la posición varía, una etapa durante la cual se sostiene en una cierta posición y otra en la que se regresa al estado inicial del músculo. Durante estas etapas se pueden medir pequeñas variaciones en la intensidad eléctrica sobre el músculo particular a revisar, y que se traducen en una Serie de Tiempo de estas intensidades. Las variaciones en tiempo y forma de

estos períodos podrían indicar síntomas, degeneraciones, agotamiento, mal nutrición, etc. Por ello es importante medir no sólo el crecimiento y disminución de estos movimientos a nivel global, sino también el comportamiento de las micro-variaciones que tienen. Para ello podemos apoyarnos de las herramientas de los *Sistemas Complejos* y sus métodos de análisis, y desarrollar con ello algunos equivalentes para las Series de Tiempo.

Recordemos que las Series de Tiempo cuya distribución de probabilidad conjunta (de ella con ella misma) no varía con el tiempo se les llama *estacionarias*. Un ejemplo de estas son las que tienen media y distribución estándar móviles cero.

Exponente de Hurst

Si recordamos, una forma de descomponer una Serie de Tiempo consiste en tomar *ventanas* de tiempo de cierto tamaño v, en las que consideraremos los datos $x_1, x_2, \ldots, x_v$, como la primera ventana, $x_2, x_3, \ldots, x_{v+1}$ como la segunda ventana, en general $x_i, x_{i+1}, \ldots, x_{i+v-1}$. Si tenemos n datos, entonces el índice del último dato será $n = i + v - 1$, por lo que despejando tenemos: $i = n - v + 1$ el cual es el número de ventanas de tamaño v que puede tener nuestra Serie.

Un primer método para analizar series con técnicas de los Sistemas Complejos consiste en el llamado *Exponente de Hurst*, el cual mide esencialmente el comportamiento fractal de la varianza de los datos a diferentes tamaños de ventana. Se puede calcular matemáticamente con el siguiente procedimiento[wiki]:

1. Calcula la media:

$$m = \frac{1}{n}\sum_{i=1}^{n} x_i$$

2. Crea una serie sin media:

$$y_t = x_t - m \quad \text{para } t=1,2, \ldots, n$$

3. Calcula las sumas acumuladas de la serie anterior:

$$z_t = \sum_{i=1}^{t} y_i \quad \text{para } t=1,2, \ldots, n$$

4. Calcula el rango R para cada tamaño de ventana v:

$$R(v) = \max(z_1, z_2, \ldots, z_v) - \min(z_1, z_2, \ldots, z_v)$$

5. Calcula la desviación estándar para cada tamaño de ventana v:

$$S(v) = \sqrt{\frac{1}{v}\sum_{i=1}^{v}(x_i - m)^2}$$

6. Ajústale una recta a la función $E(v) = R(v)/S(v)$ en el plano *log-log* utilizando, por ejemplo, el método de *mínimos cuadrados*. La pendiente de esa recta será el exponente de Hurst.

Para los pasos 4-6 se pueden tomar potencias de la base del logaritmo (2 o 10, por ejemplo) como tamaños de ventana, para que en el eje x se tengan valores enteros y sean más fáciles los cálculos.

Utilizando la biblioteca Numpy de Python, el código para calcular el exponente de Hurst para la Serie de Tiempo *x* puede escribirse así:

```python
import numpy as np


def hurst(x):
    """ Para simplificar, asumamos x es del tipo
        np.array([float,float,...]) ... """
    n = x.size
    m = x.mean()  # 1
    y = x - m  # 2
    z = y.cumsum()  # 3
    nventanas = np.log2(n)
    ventanas = np.arange(1, nventanas+1)
    e = np.zeros_like(ventanas)
    i = 0
    for ven in ventanas:
        v = int(2**ven)
        r = np.max(z[:v])-np.min(z[:v])  # 4
        s = np.std(y[:v])  # 5
        e[i] = r/s
        i += 1
    return np.polyfit(ventanas, np.log2(e), 1)[0]  # 6


# Calcula el exponente de Hurst de 100000 valores aleatorios. h[i]
# Muestra valores mínimo, máximo y std de los exponentes para 1000 experimentos
h = np.zeros(1000)
for i in range(1000):
    x = np.random.rand(100000)
    h[i] = hurst(x)
print(f"min={np.min(h)}, max={np.max(h)}, s={np.std(h)}")
```

Cómo puede apreciarse, el cálculo del exponente de Hurst es muy parecido al del cálculo de la *Dimensión Fractal por Cajas* para figuras geométricas, y nos puede ayudar a determinar el grado de *auto-semejanza* de los datos de la Serie.

Del paso 6 se puede ver que el exponente de Hurst nos dirá el grado de parecido de los rangos para cada ventana, con una *Ley de Potencias*, por lo que en realidad el exponente es calculado teóricamente como una esperanza estadística.

El método del exponente de Hurst es muy utilizado, desafortunadamente su valor es poco confiable cuando la Serie de Tiempo no es estacionaria, para esos casos es preferible el método DFA.

DFA

Otro método muy utilizado para el análisis no-lineal de Series de Tiempo es el *DFA* (*Detrended Fluctuation Analysis*) *Análisis de Fluctuaciones sin Tendencia*, originalmente aplicado por Peng y otros [peng] sobre nucleótidos del ADN, y que se puede aplicar sobre Series de Tiempo de todo tipo, aunque no sean estacionarias.

Algunas ventajas de DFA sobre Hurst o Análisis Espectral son "que permite la detección de correlaciones a larga escala y también evita la detección espuria de correlaciones aparentemente a larga escala, las cuales son un artefacto de series no estacionarias." [havl]

Para calcular el DFA de la Serie de Tiempo $x_1, x_2, \ldots, x_n$, seguimos estos pasos [havl]:

1. Calcula la media:

$$m = \frac{1}{n}\sum_{i=1}^{n} x_i$$

2. Calcula y_t, las sumas acumuladas desplazando la media global, de la Serie de Tiempo:

$$y_k = \sum_{i=1}^{k} (x_i - m)$$

3. Divide y_k , para k=1,2,…n en *cajas* (ventanas disjuntas) de tamaño v, llamémoslas X_r, para r=1,2, …, n/v (por simplicidad, asumamos que n es múltiplo de v).
$X_1=(y_1, y_2, \ldots, y_v)$, $X_2=(y_{v+1}, y_{v+2}, \ldots, y_{2v})$, …, $X_{n/v}=(y_{n-v+1}, y_{n-v+2}, \ldots, y_n)$

4. Por un método de mínimos cuadrados, calcular las ordenadas de la recta que mejor aproxime los valores de cada caja X_r . Llamémos $y_v(t)$ a la ordenada para el tiempo t. Esta será la tendencia local a la caja X_r, con $r = t//v+1$. (// = división entera)

5. Quitémosle la tendencia a la suma acumulada, y calculemos su fluctuación media cuadrática de esta manera:

$$F(v) = \sqrt{\frac{1}{n}\sum_{t=1}^{n}[y_t - y_v(t)]^2}$$

6. Al igual que con el exponente de Hurst, grafica $F(v)$ en escala *log-log*, aproxímale una recta, de nuevo, por mínimos cuadrados. El DFA de la serie original será la pendiente de esta recta.

Un valor DFA de 0.5 significa que la suma acumulada proviene de una serie aleatoria. Esto es, consiste de valores completamente sin correlación. Si hay correlaciones a corto plazo, conforme crece v, crece el valor de la DFA hasta acercarse a 0.5. Si el valor es mayor que 0.5 pero menor que 1, esto indica correlaciones a largo plazo con forma de leyes de potencias. DFA=1 indica ruido 1/f, el cual es un punto medio entre ruido blanco (impredecible) y ruido Browniano (paisaje muy "rugoso"). Si por ejemplo el DFA=1.3, la serie será más Browniana.

Como comentario, este es el algoritmo de Peng, muy parecido a su versión original, aunque en versiones más recientes se habla de empalmar las cajas del paso 3 hasta en un 50%.

IFSOAC - Juego Circular del Caos

Los *Sistemas de Funciones Iteradas* (IFS) son un método ideado por *M. Barnsley*, entre otras razones, para generar fractales. En su versión mas simple es el conocido *Juego del Caos*, donde dados los vértices de un triángulo A_1, A_2, A_3 y un punto x_0 al azar, un nuevo punto x_1 es construido a la mitad de la distancia entre x_0 y uno de estos vértices escogido al azar, y así sucesivamente. Es decir, $x_{i+1} = (A_j + x_i) / 2$, donde j es escogida al azar para cada i. Esto nos produce un fractal bien conocido, el *Triángulo de Sierpinski*.

En general el *IFS* consta de un conjunto finito de funciones *conservativas* (donde su co-dominio es subconjunto del dominio), por lo general se usan transformaciones *afines*, las cuales constan cada una de un escalamiento, una rotación y una traslación, todo lo cual no involucra más que operaciones lineales, es decir, sumas y multiplicaciones. Al igual que en el Juego del Caos, los IFS tienen además un punto inicial al azar x_0, al cual se le aplica una de las funciones escogida al azar, generando x_1, y así sucesivamente. Es decir $x_{i+1} = f_j(x_i)$, donde j es escogida al azar para cada i.

En 2004, Gustavo Carreón y Jesús Enrique Hernández, realizaron una tesis bajo la dirección de Pedro Miramontes, en la que desarrollaron el Juego del Caos sobre un "círculo" (*IFSOAC = IFS On a Circle*), mas bien un polígono regular de muchos lados, es decir, se tienen como vértices iniciales $A_1, A_2, \ldots, A_m$, y para la elección de x_{i+1}, se escoge, no al azar, sino el A_j correspondiente al punto y_i de una serie de tiempo con el cual promediar x_i, eligiendo por ejemplo j, con el mapeo lineal del valor de y_i en el intervalo entero [1, m], de esta manera:

$$j = \left\lfloor \frac{y_i - y_{min}}{y_{max} - y_{min}} \cdot m \right\rfloor + 1$$

Donde y_{max}, y_{min} son los valores máximos y mínimos de la Serie de Tiempo, conservando j entera.

El resultado es una serie de arcos fractales cuya comparación con los correspondientes a Series de Tiempo distribuidas como ciertos ruidos, nos permite identificar visualmente muy rápido las características fractales de la Serie.

Referencias

[wiki] https://en.wikipedia.org/wiki/Hurst_exponent

[peng] Peng, C. K., Buldyrev, S. V., Havlin, S., Simons, M., Stanley, H. E., & Goldberger, A. L. (1994). Mosaic organization of DNA nucleotides. *Physical review e*, *49*(2), 1685. https://link.aps.org/pdf/10.1103/PhysRevE.49.1685

[havl] Peng, C. K., Havlin, S., Stanley, H. E., & Goldberger, A. L. (1995). Quantification of scaling exponents and crossover phenomena in nonstationary heartbeat time series. *Chaos: an interdisciplinary journal of nonlinear science*, *5*(1), 82-87. http://havlin.biu.ac.il/PS/Quantification%20of%20scaling%20exponents%20and%20crossover%20phenomena%20in%20nonstationary%20heartbeat%20time%20series.pdf

[carr] Carreon Vazquez, Gustavo sustentante. El juego circular del caos en el ADN y compresion fractal de imagenes / 2004. http://oreon.dgbiblio.unam.mx/F/TM2K562RPLJ5E8RILHKX6XG8S4K2B39KBUE2H89Y89GM5IY4I7-10536?func=full-set-set&set_number=005818&set_entry=000003&format=999