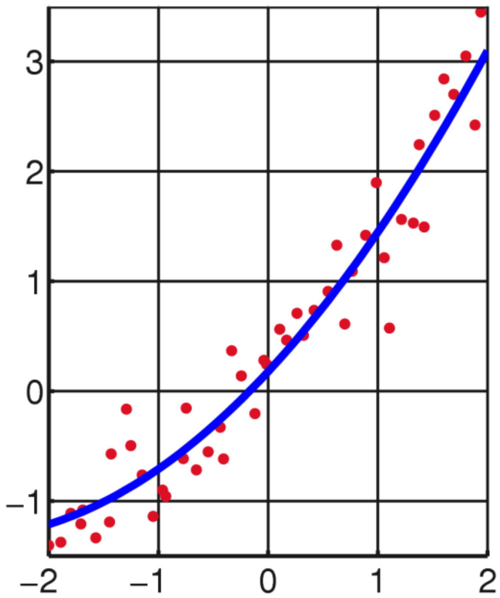
Ajuste de funciones por mínimos cuadrados (funciones no polinómicas)

Procedimiento general



Hay situaciones en las que tenemos una serie o **nube de puntos** de los cuales queremos obtener una **función** del tipo que elijamos, que se aproxime a ellos lo máximo posible, con la finalidad de poder predecir el valor de esa función para cualquier otro valor de la variable independiente.

Para obtener esa función, utilizamos la técnica de **aproximación por mínimos cuadrados**, la cual consiste en minimizar el cuadrado de la distancia entre cada punto y la función a obtener.

¿En qué consiste la técnica de **aproximación por mínimos cuadrados?**

Se resume en 4 sencillos pasos:

1. Hallar expresión de distancia entre función y cada punto (dj), y elevarla al cuadrado.
2. Derivar con respecto a cada variable e igualar a 0.
3. Plantear el sistema de ecuaciones.
4. Resolver el sistema o elaborar un algoritmo que lo resuelva.

Explicación proceso:

1. Queremos la mínima distancia posible entre los puntos y la función a obtener (**y(t)**), por lo que tenemos que hallar el mínimo de la función **f** que expresa el cuadrado de esa distancia (se utiliza al cuadrado debido a que si la distancia es pequeña la hace aún más pequeña). Para ello, debemos encontrar los puntos en los que su derivada es 0 (punto crítico).
2. Como normalmente vamos a obtener una expresión con más de una variable, debemos derivar con respecto a cada variable e igualar a 0
3. Posteriormente, deberemos unir todas las derivadas parciales igualadas a 0 en un mismo sistema de ecuaciones, para obtener el valor de cada variable.
4. El último paso es resolver el sistema. Se podrá hacer manualmente si el ejercicio nos proporciona el valor de los puntos (coordenadas x e y); en otro caso, se elaboraría un algoritmo que fuera capaz de resolver ese sistema para un número de puntos cualquiera.

Con este procedimiento podemos aproximar una nube de puntos a una función de cualquier tipo (**lo elegimos nosotros**), como por ejemplo una polinómica, logarítmica o potencial. En este caso, trabajaremos con una función no polinómica, en concreto una racional del tipo a + $b(x+1)^{-1}$, que también puede ser expresada como a + b/(x+1).

Ajuste de una función racional del tipo a + b/(x+1)

Se desea ajustar la función $\boxed{y(t) = a + \frac{b}{t+1}}$

a n puntos de coordenadas (t_i,y_i), i = 1,…,n; utilizando el criterio de mínimos cuadrados.

1. Se debe minimizar la distancia entre los puntos y la función a obtener. Esa distancia la expresaremos como f (a,b), y será la función a minimizar, de la que obtendremos los valores **a** y **b**.

$$d_i = y_i - (a + \frac{b}{t+1}) \longrightarrow d_i^2 = (y_i - (a + \frac{b}{t+1}))^2 \longrightarrow f(a,b) = \sum_{i=1}^{n}\left(yi - \left(a + \frac{b}{t+1}\right)\right)^2$$

2. Ahora, para minimizar la función, derivamos con respecto a las variables a y b:

$$\frac{\partial f}{\partial a} = 2\sum_{i=1}^{n}\left(yi - \left(a + \frac{b}{ti+1}\right)(-1)\right) = 0$$

$$\frac{\partial f}{\partial b} = 2\sum_{i=1}^{n}\left(yi - \left(a + \frac{b}{ti+1}\right)(-\frac{1}{t+1})\right) = 0$$

3. De aquí obtenemos el siguiente sistema de ecuaciones,

$$\left.\begin{array}{l} -\sum_{i=1}^{n} y_i + \sum_{i=1}^{n} a + \sum_{i=1}^{n}\frac{b}{t_i+1} = 0 \\ -\sum_{i=1}^{n} y_i\frac{1}{t_i+1} + \sum_{i=1}^{n}\frac{a}{t_i+1} + \sum_{i=1}^{n}\frac{b}{(t_i+1)^2} = 0 \end{array}\right\} \quad \left.\begin{array}{l} an + b\sum_{i=1}^{n}\frac{1}{t_i+1} = \sum_{i=1}^{n} y_i \\ a\sum_{i=1}^{n}\frac{1}{t_i+1} + b\sum_{i=1}^{n}\frac{1}{(t_i+1)^2} = \sum_{i=1}^{n}\frac{y_i}{(t_i+1)} \end{array}\right\}$$

cuya expresión matricial es

$$\boxed{\begin{pmatrix} n & \sum_{i=1}^{n}\frac{1}{t_i+1} \\ \sum_{i=1}^{n}\frac{1}{t_i+1} & \sum_{i=1}^{n}\frac{1}{(t_i+1)^2} \end{pmatrix} \times \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n}\frac{y_i}{(t_i+1)} \end{pmatrix}}$$

A X V

A la hora de realizar el algoritmo, debemos pensar en cómo representar un elemento genérico de la matriz A y del vector V.

· **En el caso de la matriz A**, observamos que cada elemento es el sumatorio desde i=1 hasta n de la expresión $\frac{1}{(t_i+1)^{(j+k-2)}}$, (siendo k la fila y j la columna), puesto que, si observamos el patrón, el exponente de la suma del denominador es igual a la suma de filas y columnas menos dos unidades.

· **En cuanto al vector V**, su término general es el sumatorio desde i=1 hasta n de la expresión $\frac{y_i}{(t_i+1)^{(k-1)}}$, (siendo k la fila).

4. El algoritmo final será el siguiente: