

P1.- Hoy en día se ha puesto de moda una rama de la inteligencia artificial que es la minería de datos. Esta se dedica a analizar el comportamiento de los distintos datos y tratar de modelar una función que explique su comportamiento.

Uno de los primeros ajuste de funciones que existe es lo que se llama una regresión lineal, que consiste en modelar los datos como si fueran una recta, aunque eso no sea necesariamente así.

Supongamos que tenemos un conjunto de vectores $\{v_1, v_2, \dots, v_n\}$ no necesariamente colineales asociados cada uno a los reales $\{t_1, t_2, \dots, t_n\}$. Nos gustaría encontrar dos vectores p y d tales que cumplan la siguiente relación:

$$v_i = p + t_i \cdot d, i = 1, \dots, n$$

Como los vectores no son colineales, es imposible encontrar p y d que cumplan la ecuación anterior, pero si es posible encontrarlos de tal manera que sea una buena aproximación.

Para encontrarlos se construyen los siguientes sistemas de ecuaciones:

$$(1) \begin{cases} v_1^1 = p_1 + t_1 d_1 \\ v_2^1 = p_1 + t_2 d_1 \\ \vdots \\ v_n^1 = p_1 + t_n d_1 \end{cases} \quad (2) \begin{cases} v_1^2 = p_2 + t_1 d_2 \\ v_2^2 = p_2 + t_2 d_2 \\ \vdots \\ v_n^2 = p_2 + t_n d_2 \end{cases} \quad (3) \begin{cases} v_1^3 = p_3 + t_1 d_3 \\ v_2^3 = p_3 + t_2 d_3 \\ \vdots \\ v_n^3 = p_3 + t_n d_3 \end{cases}$$

Donde $v_i = (v_i^1, v_i^2, v_i^3)^T$, $p = (p_1, p_2, p_3)^T$ y $d = (d_1, d_2, d_3)^T$. Estos sistemas se pueden representar matricialmente como:

$$\begin{bmatrix} 1 & t_1 \\ 1 & t_2 \\ \vdots & \vdots \\ 1 & t_n \end{bmatrix} \cdot \begin{bmatrix} p_j \\ d_j \end{bmatrix} = \begin{bmatrix} v_1^j \\ \vdots \\ v_n^j \end{bmatrix}, j = 1, 2, 3.$$

Este sistema es esencialmente incompatible, pero gracias a la función `linalg.lstsq` (visto en guía 3) se puede llegar a una solución que minimiza el error.

El archivo **Vectores.csv** contiene una serie de vectores asociados a los tiempos incluidos en el archivo **tiempo.csv**. Encuentre los vectores p y d que definen la recta que mejor se ajustan a estos vectores y grafique la recta en cuestión. Además en el mismo gráfico incluya los vectores, pero como puntos, de tal manera que se pueda ver la relación en los vectores y la recta encontrada.

Observación: Para graficar como puntos puede usar `ax.plot3D(X,Y,Z, 'or')`.