Proyecto 4

Ruben Cuadra

Dr. Victor de la Cuva

Aprendizaje Automatico

September 4, 2017

Regresión logística

Implementación en python de regresión logística la cual sucede cuando los datos se aproximan por medio de una hiperrecta (polinomio lineal) dada por la hipótesis :

$$h(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots \theta_n x_n = \theta^T X$$

Donde el conjunto de datos X es = $[x_1, x_2, \ldots x_n]^t$ y usando la función sigmoidal. $g(z) = \frac{1}{1 + e^{-z}}$, recordando que $z = h(x)$

El código consta de una librería llamada Proyecto4.py la cual consta de 9 funciones, todas giran entorno a la función: (Las negritas son los parámetros que recibe)

aprende:

theta : Matrix de valores en theta, puede ser *None*
X : Matrix de valores en X
Y : Vector con valores en Y, mismo tamaño de X
iteraciones (opcional): Default es 100

Nos devuelve 1 vector con N lugares que representa los valores thetas encontrados, N = cantidad de filas en X, se itera sobre la ecuación

$$\theta_j = \theta_j - \frac{1}{n} \sum_{i=0}^{n} (g(h(x_i)) - y_i) x_i^j$$

*n es la cantidad de datos que tenemos(Renglones en la matriz X o Y)

Ejecuta esa formula *iteraciones* veces y obtiene el costo

graficarDatos:

X : Matriz con valores en X
Y : Vector con valores en Y
theta : Vector obtenido previamente usando la función aprende.

Graficas los datos en X usando como eje vertical representa la segunda columna de la matriz, el eje horizontal es la primer columna de la matriz

funcionCosto:

X : Matrix de valores en X
Y : Vector con valores en Y
thetas : Vector con valores theta

Regresa un valor numérico obtenido de la ecuación

$$J(\theta) = \frac{1}{n}\sum_{i=0}^{n} - y_i log(g(h(x_i))) - (1 - y_i) log(1 - g(h(x_i))$$

ecuacionNormal:

X : Matrix de valores en X
Y : Vector con valores en Y

Regresa una matriz con los valores theta usando la ecuación:

$$\theta = (X^T X)^{-1} X^T Y$$

normalizacionDeCaracteristicas:

X : Matrix de valores en X

Regresa una tupla con 3 valores
_X = Matriz normalizada
μ = Vector con la media para cada columna de la matriz X
σ = Vector con desviaciones estándares para cada columna

La normalisation de la matriz se da por la formula

$$x_i = \frac{x_i - \mu}{\sigma}$$

predice:

X : Matriz de valores X

Theta : Vector de valores en Theta

Regresa un vector p que contiene 1 o 0 dependiendo si esas X evaluadas con la función sigmoidal el resultado es superior a 0.5 o inferior

Requisitos

Libreria matplotlib (Graficas)

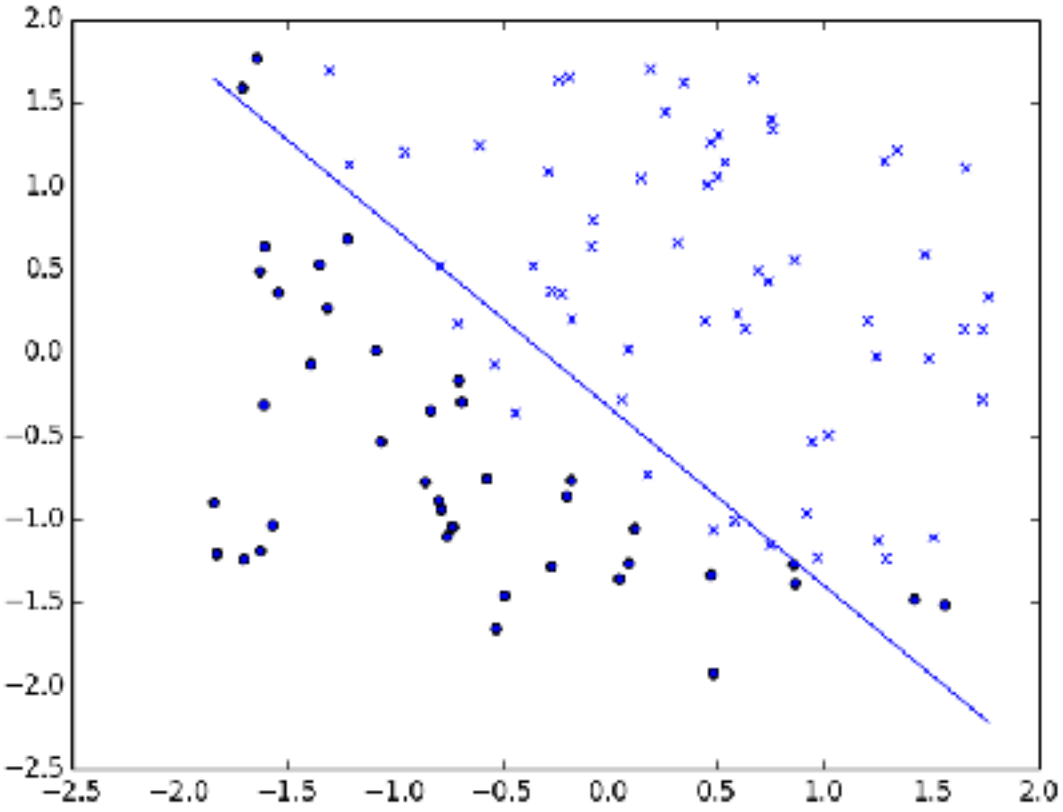
Libreria numpy (Operaciones matemáticas)

Python 2.7 o 3.5 (Instalar las librerías correctamente usando pip o pip3)

En el repositorio se encuentra un archivo ejemplo que es parseado por la funcion *getDataFromFile* la cual recibe como parámetro un archivo(su ruta absoluta) y nos devuelve la matriz X y el vector Y, los cuales ya pueden ser usados para todas las funciones previamente descritas **datos.csv** contiene valores numéricos x,y ; Son N columnas separadas por comas donde la ultima representara las Y y las demás valores en X.

Tras correr una serie de pruebas con la información ejemplo se llego a la conclusión de obtener la información del archivo usando **getDataFromFile** (Ya tendremos X y Y)posteriormente usaremos **normalizacionDeCaracteristicas** con el resultado en X de la función anterior, una vez que obtenemos una X normalizada se manda a llamar la función **gradienteDescendenteMultivariable** con la X normalizada, el vector Y, y una None en la variable thetas con *iteraciones* a 1500 debemos obtener un arreglo **thetas: [1.71835438 3.99258451 3.72493998]** (Que representan theta0,theta1,theta2) Que al ser pasadas como parámetro a *funcionCosto* debemos obtener como **costo**

0.2035, al traficar los datos y la hiperrecta generada por las thetas obtenemos una imagen muy parecida a: (Todo este proceso se puede visualizar en el archivo main.py)



La recta representa a los alumnos que sus resultados arrojaban un resultado de 0.5 en la sigmoidal, de modo que podemos predecir que cualquier valor por debajo de esta misma representa que el alumno no sera admitido y cualquier valor por encima representa que pasara, podemos ver que es bastante acertado con los usuarios que pasaron(x) vs los que no (o). No obstante tal vez seria mejor una función de mayor grado, una linea un poco mas curveada.