Regresión Logstica y Descenso por Gradiente

José Castro

1 Introducción

En este ejercicio usted trabajará con regresión logística con dos conjuntos de datos. Para empezar debe bajar el código que se encuentra en la página del curso en el tecDigital.

2 Archivos en esta tarea

Los archivos incluidos en esta tarea son:

- `ex2.m` Un script de Octave que le puede ayudar a seguir esta tarea.
- `ex2_reg.m` Otro script para la segunda parte de esta tarea.
- `ex2data1.txt` datos de entrenamiento para la primera parte de la tarea.
- `ex2data2.txt` datos de entrenamiento para la segunda parte de la tarea.
- `mapeCaracteristicas.m` función para generar características polinomiales.
- `grafiqueFrontera.m` función para graficar la frontera de decisión,
- `[*] grafiqueDatos.m` función para graficar los datos en 2D.
- `[*] sigmoide.m` función sigmoide.
- `[*] funcionDeCosto.m` función de costo de regresión logística.

- `[*] prediccion.m` función para predecir utilizando el modelo.
- `[*] funcionDeCostoReg.m` función de costo con regularización.

* indica un archivo que usted debe completar.

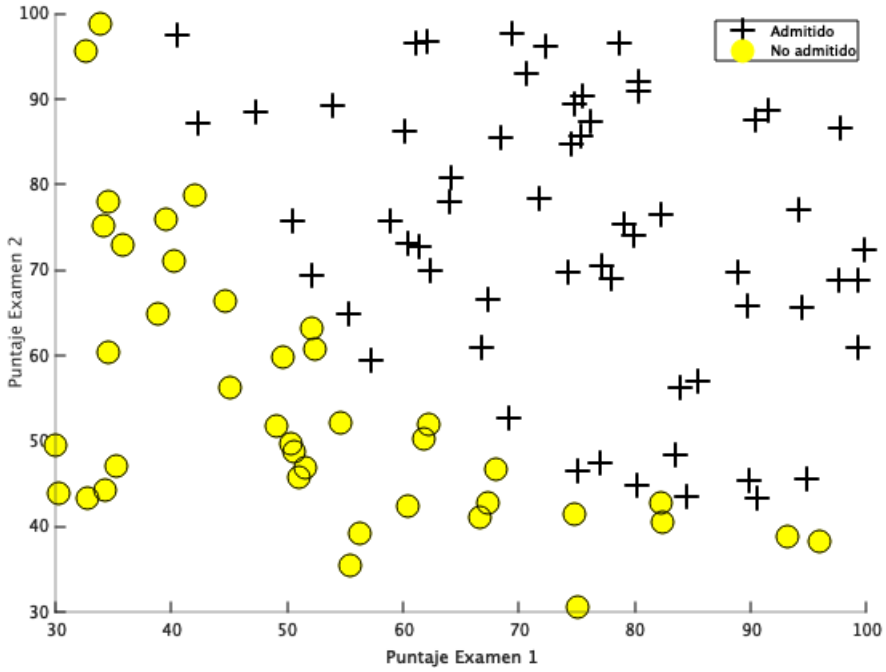
A través de este ejercicio usted estará utilizando los scripts `ex2.m` y `ex2_reg.m`. Estos preparan los datos para los problemas y llaman a las funciones que usted escribirá, no necesita modificarlos. Solo se requiere que modifique las funciones en otros archivos (los que tienen *).

3 Regresión logística

En esta sección construirá un modelo de regresión logística para predecir si un estudiante ingresa a una universidad. Su tarea es construir un modelo de clasificación que estime la probabilidad de admisión basado en el resultado de dos exámenes. Este archivo y el código en `ex2.m` le guiará en este ejercicio.

3.1 Visualizar los Datos

Antes de empezar siempre es bueno visualizarlos si esto es posible. En la primera parte de ex2.m el código carga los datos y los debe desplegar en un gráfico de 2 dimensiones. Usted debe completar `grafiqueDatos.m` para que despliegue una figura como la



3.1.1 Implementación

3.1.2 función sigmoide

Antes de empezar debe implementar la función que calcula la hipótesis utilizando regresión logística:

$$h_\theta(x) = g\left(\theta^T x\right)$$

donde g es la función sigmoide definida como:

$$g(z) = \frac{1}{1 + e^{-z}}$$

su primer tarea es implentar esta función en el archivo `sigmoide.m`. Esta función debe funcionar tanto para un escalar como para un vector o una matriz, en cuyo caso calcula $g(z)$ para cada uno de los elementos de la matrix/vector.

3.1.3 función de costo y gradiente

Ahora debe impkementar la función de costo en el archivo `funcionDeCosto.m` el cual debe calcular tanto el costo como el gradiente.

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} x_j^{(i)} \right)$$

y el gradiente del costo es un vector del mismo largo que θ donde el j-ésimo elemento (para $j = 0, 1, \ldots, n$) está definido como:

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right) x_j^{(i)}$$

Note que este gradiente se ve indéntico al de regresión lineal, pero realmente es distinto debido a que la definición de $h_\theta(x)$ es distinta.

Una vez que termina esto `ex2.m` llamará a `funcionDeCosto.m` utilizando los parámetros iniciales de θ. Esto debe dar un valor cercano a 0.693.

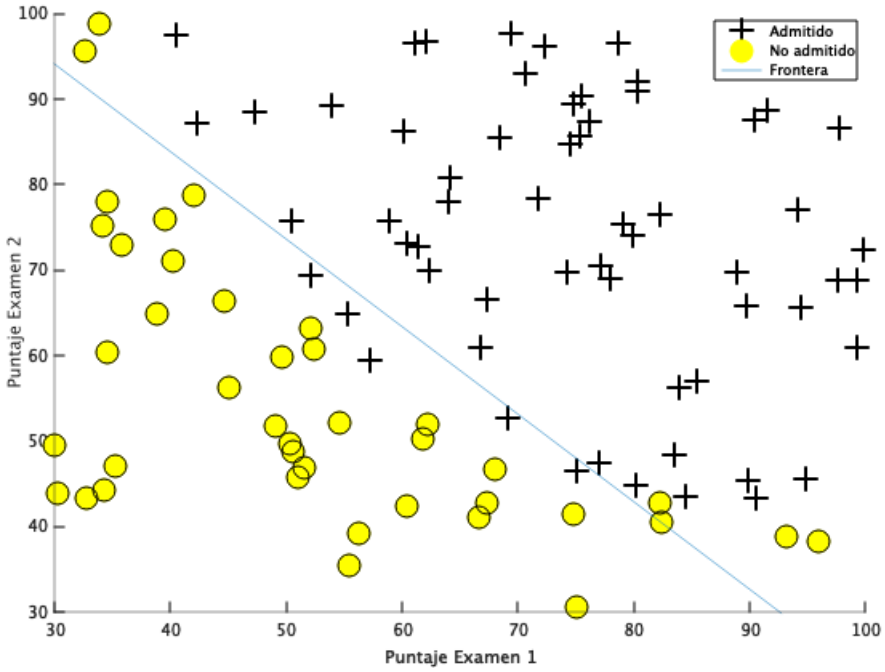
3.1.4 Parámetros de aprendizaje utilizando `fminunc`

En la tarea anterior le toco encontrar parámetros óptimos para el modelo de regresión lineal por descenso por gradiente. Esta vez, en vez de iterar a pie se utiliza la función de optimización de Octave `fminunc`

La función de Octave `fminunc` encuentra el mínimo de una funcón. Para regresión logística debe optimizar la función de costo $J(\theta)$ con los parámetros θ.

Usted debe crear la función de costos en `funcionDeCosto.m`, si todo sale bien `ex2.m` llamará a su función de costos y el resultado debe ser cercano a 0.203.

Con el valor final de θ se grafica la frontera de decisión, y usted debe poder ver una figura similar a la siguiente:



3.1.5 Evaluando a la regresión logística

Después de aprender los parámetros, se utilizará el modelo para predecir la probabilidad que un determinado estudiante sea admitido. Para un estudiante con 45 en el promer examen y 85 en el segundo, usted debe esperar una probabilidad de ser admitido cercana a 0.776.

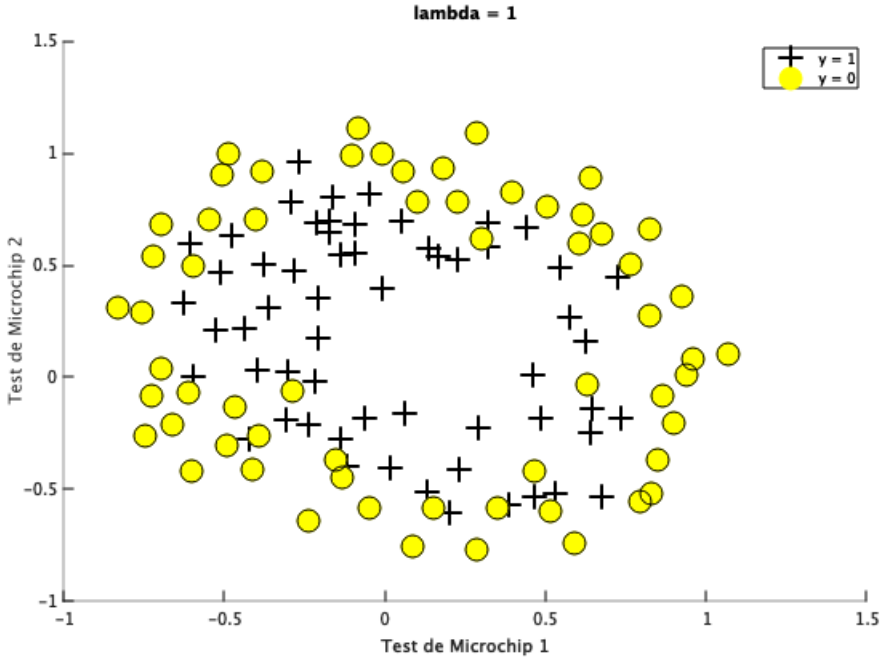
En esta parte de la tarea entonces, usted debe completar la función `prediccion.m`.

4 Regresión logística regularizada

En esta parte usted implementará regresión logística con una modificación que ayuda que esta no sobre-estime los parámetros. Aquí los datos corresponden a microchips que se prueban para asegurarse que funcionan correctamente. En esta parte utilizará el archivo `ex2_reg.m`.

4.1 Visualizando los datos

En esta sección, de manera similar a el anterios, se utiliza `grafiqueDatos.m` para visualizar el set de entrenamiento. Debe ver algo como lo siguiente:



Estos datos claramente no se pueden separar utilizando características lineales.

4.2 Mapeo de características

Una manera de mejorar el modelo es utilizando combinaciones de características, la función en el archivo `mapeeCaracteristicas.m` le genera car-

acterísticas hasta la sexta potencia:

$$\text{mapeeCaracteristicas(x)} = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ x_1^2 \\ x_1 x_2 \\ x_2^2 \\ x_1^3 \\ \vdots \\ x_1 x_2^5 \\ x_2^6 \end{bmatrix}$$

Como resultado de este mapeo de dos características (los resultados de los dos exámenes) las características se han transformado a un vector de 28-dimensiones (28 características). Aunque las nuevas características pueden generar un clasificador más expresivo, también lo hace más susceptible a sobre-ajustar los datos (overfitting). En este ejercicio utilizaremos el concepto de regularización para evitar esto.

4.3 Función de costo y gradiente

Ahora implementara código para hacer la nueva función de costo y gradiente en el archivo `funcionDeCostoReg.m`, los valores para regresión loística regularizada son:

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \left[-y^{(i)} \log(h_\theta(x^{(i)})) - (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)})) \right] + \frac{\lambda}{2m} \sum_{j=1}^{n} \theta_j^2$$

Esta función de costo es igual a la anterior solo que ahora se agrega la sumatoria del cuadrado de los θ_j multiplicado por una constante λ la cual es un parámetro del algoritmo. Note que la sumatoria empieza de 1, así que θ_0 no es parte de ella, ahora bien, como en Octave los vectores se indexan empezando en 1, esto significa que debe utilizar los valores de la variable `theta` a partir del 2 en adelante (`theta(1)` contiene a θ_0).

El gradiente de θ se define como:

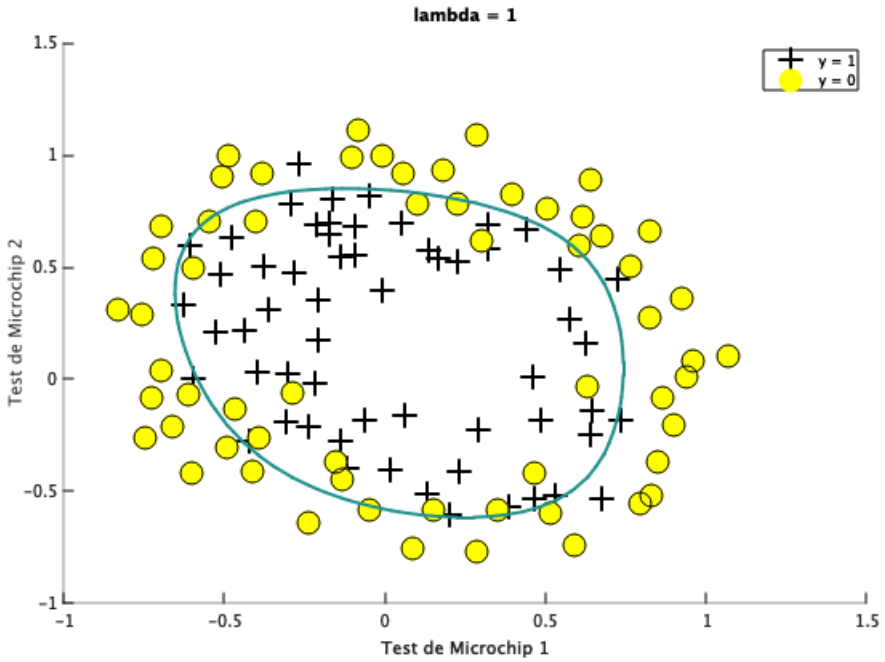
$$\frac{\partial J(\theta)}{\partial \theta_0} = \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right) x_j^{(i)} \qquad \text{para } j = 0$$

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right) x_j^{(i)} + \frac{\lambda}{m} \theta_j \quad \text{para } j \geq 1$$

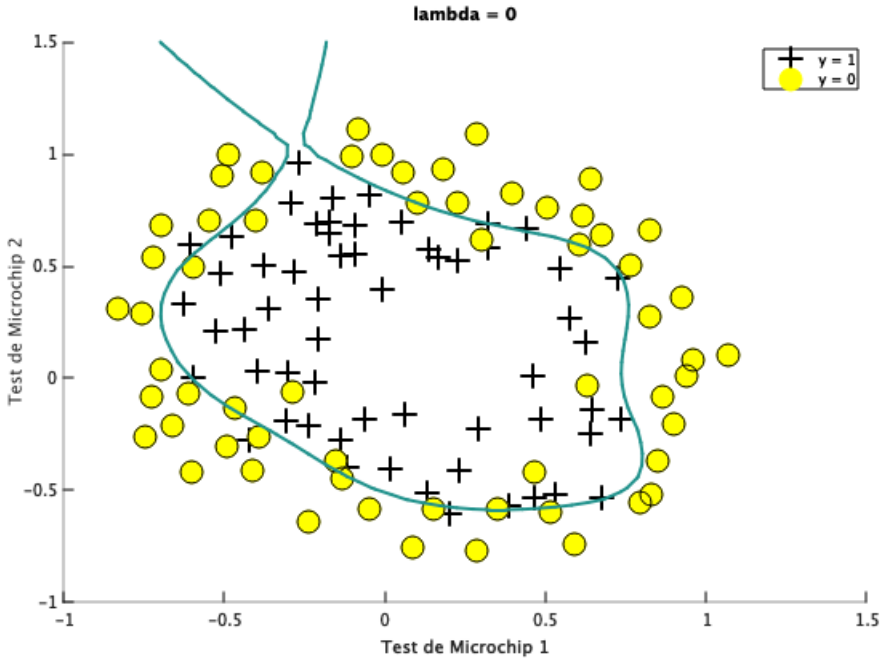
Una vez que termine, `ex2_reg.m` llamará a su función de costo con el valor inicial de θ (todos 0's). Debe observar un resultado cercano a 0.693.

4.4 Graficando la frontera de decisión

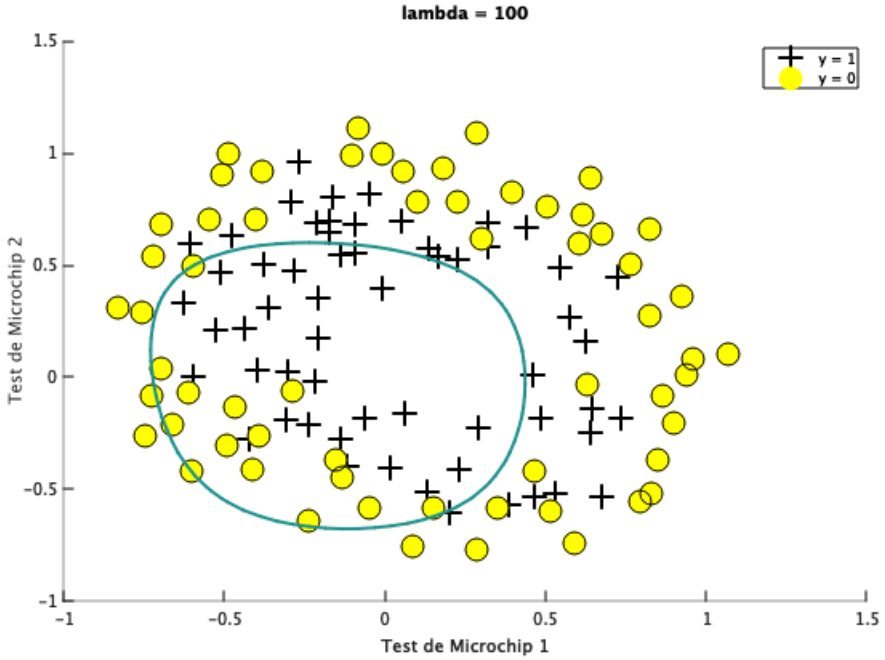
Una ves hecho `funcionDeCostoReg.m` el archivo `ex2_reg.m` graficará la frontera de decisión y podrá ver algo como el grafico siguiete:



Debe probar el parendizaje cambiando el valor de lambda para ver como afecta esto a la frontera de decisión. Cuanto más grande sea λ, más simple será la frontera de decisión. Debe poder generar gráficos como los siguientes cambiando el valor de lambda:



En cuyo caso esta sobre ajustando los datos, y como:



5 Evaluación

Todos los puntos de la tarea valen igual.