Regresión Logística

Sebastián Marroquín Martínez

Noviembre, 2018

1. ¿Qué es la Regresión Logística?

La regresión logística es en realidad un método de clasificación. Este tipo de método introduce una no linealidad adicional sobre un clasificador lineal, esto es: $f(x) = w^T x + b$, usando una función logística (o sigmoide), $\sigma()$.

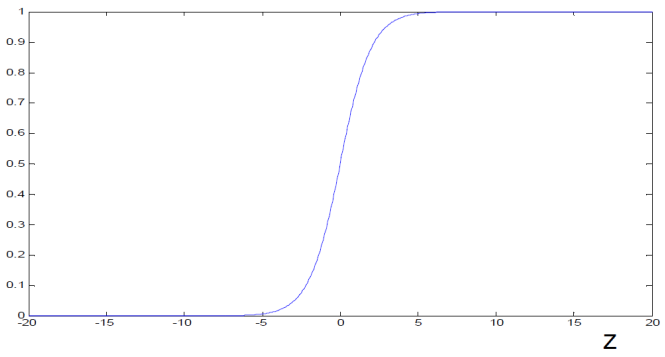
El clasificador de la Regresión Logística se define a continuación:

$$\sigma(f(x_i)) = \begin{cases} \geq 0{,}5 y_i = +1 \\ < 0{,}5 y_i = -1 \end{cases}$$

Donde, $\sigma(f(x)) = \frac{1}{1+\exp^{-f(x)}}$.

La función logística o función sigmoide se caracteriza por lo siguiente:

$\sigma(z)$ $=$ $\frac{1}{1+\exp^{-z}}$



1. A medida que z va de infinito negativo a infinito positivo , la sigmoide va de 0 a 1, una "función de aplastamiento".

La función sigmoide calcula el valor de $(w^T x)$ entre $(0, 1)$, para retornar el valor en una probabilidad. Por lo que tenemos lo siguiente:

$$P(y = 1|x, w) = \mu = \sigma(w^T x) = \frac{1}{1 + e^{-w^T x}} = \frac{e^{w^T x}}{1 + e^{w^T x}} \tag{1}$$

$$P(y = 0|x, w) = 1 - \mu = 1 - \sigma(w^T x) = \frac{1}{a + e^{w^T x}} \tag{2}$$

1.1. Límite de Decisión

En el límite de decisión, ambas clases son equiparables (1, 2). Así:

$$\begin{aligned} P(y = 1|x, w) &= P(y = 0|x, w) \\ \frac{e^{w^T x}}{1 + e^{w^T x}} &= \frac{1}{1 + e^{w^T x}} \\ e^{w^T x} &= 1 \\ w^T x &= 0 \end{aligned}$$

Por lo tanto, si $y = 1$, sí $w^T x \geq 0$, de lo contrario $y = 0$.

Renombremos como sigue:

$$P(y = 1|x, w) = \mu = \frac{1}{1 + e^{-(w^T x)}} \tag{3}$$

Por lo que:

1. High Positive Score to $w^T x$: $\left\{ \right.$ Alta probabilidad del label = 1
2. High Negative Score to $w^T x$: $\left\{ \right.$ Probabilidad baja del label = 1. (Alta probabilidad de label = 0).

1.2. Estimación de Parámetros

Cada etiqueta y_n es binaria, con una probabilidad μ_n.

$$P(y|x,w) = \prod_{n=1}^{N} P(y_n|x_n,w) = \prod_{n=1}^{N} \mu_n^{y_n}(1-\mu_n)^{1-y_n} \tag{4}$$

Donde $\mu = \frac{e^{w^T x_n}}{1+e^{w^T x_n}}$

2. Función de Costo

Si usamos el error cuadratico como una función de costo para la regresión logística $J(w)$, será no convexa.

Por lo que utilizaremos:

$$Cost(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) if y = 1 \\ -\log(1-h_\theta(x)) if y = 0 \end{cases}$$

Juntando todo, tenemos que:

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} Cost(h_\theta(x^i), y^i) \tag{5}$$

$$= -\frac{1}{m}[\sum_{i=1}^{m} y^i \log(h_\theta(x^i) + (1-y^i)\log(1-h_\theta(x^i)))] \tag{6}$$

La regla de actualización del gradiente nos queda de la siguiente manera:

$$\theta_j := \theta_j - \alpha \sum_{i=1}^{m} (h_\theta(x^i) - y^i) \cdot x_j^i \tag{7}$$

2.1. Derivada de la Función de Costo

La razón es la siguiente. Usamos la notación:

$$\theta x^i := \theta_0 + \theta_1 x_1^i + \cdots + \theta_p x_p^i.$$

La derivada de la función Sigmoide es la siguiente:

$$\begin{aligned}
\frac{d}{dx}\sigma(x) &= \frac{d}{dx}\left(\frac{1}{1+e^{-x}}\right) \\
&= \frac{-(1+e^{-x})'}{(1+e^{-x})^2} \\
&= \frac{e^{-x}}{(1+e^{-x})^2} \\
&= \left(\frac{1}{1+e^{-x}}\right)\left(\frac{e^{-x}}{1+e^{-x}}\right) \\
&= \left(\frac{1}{1+e^{-x}}\right)\left(\frac{1+e^{-x}}{1+e^{-x}} - \frac{1}{1+e^{-x}}\right) \\
&= \sigma(x)\left(\frac{1+e^{-x}}{1+e^{-x}} - \sigma(x)\right) \\
&= \sigma(x)\left(1-\sigma(x)\right)
\end{aligned}$$

Derivada de la Funcion de Costo:

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

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

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

$$= \frac{-1}{m} \sum_{i=1}^{m} \left[y^{(i)} \frac{\frac{\partial}{\partial \theta_j} \sigma \left(\theta^\top x^{(i)} \right)}{h_\theta \left(x^{(i)} \right)} + (1 - y^{(i)}) \frac{\frac{\partial}{\partial \theta_j} \left(1 - \sigma \left(\theta^\top x^{(i)} \right) \right)}{1 - h_\theta \left(x^{(i)} \right)} \right]$$

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

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

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

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

3. Regularización

El ajuste excesivo es un problema común en el aprendizaje automático, donde un modelo se desempeña bien en los datos de entrenamiento, pero no se generaliza bien en datos invisibles (datos de prueba).

Si un modelo sufre un ajuste excesivo, también decimos que el modelo tiene una gran varianza, lo que puede deberse a que haya demasiados parámetros que conduzcan a un modelo demasiado complejo dados los datos subyacentes. De manera similar, nuestro modelo también puede sufrir (alto sesgo), lo que significa que nuestro modelo no es lo suficientemente complejo como para capturar bien el patrón en los datos de entrenamiento y, por lo tanto, también tiene un bajo rendimiento en datos invisibles.

Una forma de encontrar una buena compensación de sesgo-variación es ajustar la complejidad del modelo a través de la regularización. La regularización es un método muy útil para manejar la colinealidad (alta correlación entre características), filtrar el ruido de los datos y, eventualmente, evitar el sobreajuste.

El concepto detrás de la regularización es introducir información adicional (sesgo) para penalizar los pesos de parámetros extremos. La forma más común de regularización es la llamada regularización de L2 (a veces también llamada contracción de L2 o disminución de peso), que se puede escribir de la siguiente manera:

$$\frac{\lambda}{2}||w||^2 = \frac{\lambda}{2}\sum_{j=1}^{m} w_j^2$$

Aquí, λ es llamado "Parametro de Regularización".

Para aplicar la regularización, solo necesitamos agregar el término de regularización a la función de costo que definimos para la regresión logística para reducir los pesos:

$$J(w) = [\sum_{i=1}^{n}(log(g(z^i)) + (1 - y^i)(-log(1 - g(z^i))))] + \frac{\lambda}{2}||w||^2$$

A través del parámetro de regularización λ, podemos controlar qué tan bien ajustamos los datos de entrenamiento mientras mantenemos los pesos pequeños. Al aumentar el valor de λ, aumentamos la fuerza de regularización.

4. Implementación

4.1. Función Sigmoide

Antes de comenzar con la función de costo real, recordemos que la hipótesis de regresión logística se define como:

$$h_\theta(x) = g(\theta^T x),$$

donde la función g is la función Sigmoide. La función sigmoidea se define como:

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

Por lo que la función sigmoide está codificado de la siguiente manera:

```
function g = sigmoid(z)
        g = zeros(size(z));
        g = 1 ./ (1+e.^-z);
end
```

4.2. Función de Costo y Gradiente

Recordemos que la función de costo en regresión logística es:

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} [-y^i \log(h_\theta(x^i)) - (1 - y^i) \log(1 - h_\theta(x^i))]$$

y el gradiente del costo es un vector de la misma longitud que θ donde el j^{th} elemento ($paraj = 0, 1, 2, ..., n$) se define como lo siguiente:

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

Debemos de tener en cuenta que si bien este gradiente tiene un aspecto idéntico al de la regresión lineal, la fórmula es realmente diferente porque la regresión lineal y logística tiene diferentes definiciones de $h_\theta(x)$.

```
function [J, grad] = costFunction(theta, X, y)
        % Calcula el costo y el gradiente para
        % la regresion logistica

        % Inicializamos algunos valores
        m = length(y); % numero de muestras de ejemplo

        % Deberemos de regresar las variables
        % de manera correcta
        J = 0;
        grad = zeros(size(theta));

        h = sigmoid(X*theta);
        J = (1/m)*(-y'* log(h) - (1 - y)'* log(1-h));
        grad = (1/m)*X'*(h - y);
end
```

4.2.1. Regresión Logística Regularizada

Recordemos que la función de costo regularizado en regresión logística es:

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

Debemos de tener en cuenta que no se debe regularizar el parámetro θ_0.

El gradiente de la función de costo es un vector donde el j^{th} elemento se define lo siguiente:

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^i) - y^i)x_j^i, paraj = 0$$

$$\frac{\partial J(\theta)}{\partial \theta_j} = (\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^i) - y^i)x_j^i) + \frac{\lambda}{m}\theta_j, paraj \geq 1$$

```
function [J, grad] = costFunctionReg(theta, X, y, lambda)
%COSTFUNCTIONREG Calcula el costo y el gradiente
%para la regresion logistica
%Usando regularizacion

%Inicializamos algunos valores
m = length(y); %numero de las muesras de prueba

%Regresaremos las variables de manera correcta
J = 0;
grad = zeros(size(theta));

%Calculamos las derivadas parciales
%para nuestra funcion de costo
%Pero el caso de nuestra funcion
%debe de ser regularizada.

h=sigmoid(X * theta);
cost = sum(-y.* log(h) -(1-y) .* log(1-h));
grad = X' * (h-y);

grad_reg = lambda * theta;
grad_reg(1) = 0;

grad = grad + grad_reg;
```

```
    %Actualizacion de J(theta)
    J = cost / m + (lambda / (2.0 * m)) * \
                sum(theta(2:size(theta)) .^ 2);
    grad = grad / m;

end
```

4.3. Mapeo de características

Una forma de mejorar los datos es crear más funciones desde cada punto de datos. Con la función de *mapFeature.m* mapearemos las características en todos los términos polinomiales de x_1 y x_2 hasta la sexta potencia.

$$mapFeature(x) = \begin{bmatrix} \\ 1 \\ x_1 \\ x_2 \\ x_1^2 \\ x_1x_2 \\ x_2^2 \\ x_1^3 \\ \dots \\ x_1x_2^5 \\ x_2^6 \end{bmatrix}$$

```
function out = mapFeature(X1, X2)
        degree = 6;
        out = ones(size(X1(:,1)));
        for i = 1:degree
                for j = 0:i
                        out(:, end+1) = (X1.^(i-j)).*(X2.^j);
                end
        end
end
```

4.4. Trazando la frontera de decisión

Para ayudarnos a visualizar el modelo aprendido por este clasificador, construiremos la función *plotDecisionBoundary.m* que traza el límite de decisión (no lineal) que separa los ejemplos positivos y negativos.

En *plotDecisionBoundary.m*, trazamos el límite de decisión no lineal calculando las predicciones del clasificador en una cuadrícula espaciada uniformemente y luego dibujamos un gráfico de contorno de dónde cambian las predicciones de $y = 0$ a $y = 1$.

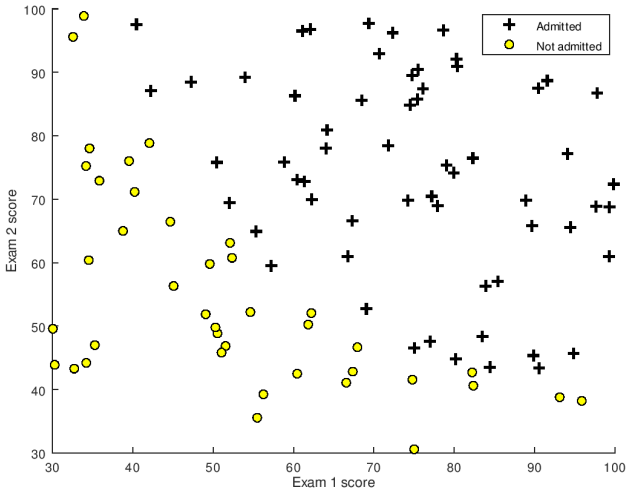
```
function plotDecisionBoundary(theta, X, y)
        %Cargamos la informacion en el plot
        plotData(X(:,2:3), y);
        hold on
        if size(X, 2) <= 3
        plot_x = [min(X(:,2))-2,  max(X(:,2))+2];
        plot_y = (-1./theta(3)).*(theta(2).*plot_x + theta(1));
        plot(plot_x, plot_y)

        legend('Admitted', 'Not admitted', 'Decision Boundary')
        axis([30, 100, 30, 100])
        else
        u = linspace(-1, 1.5, 50);
        v = linspace(-1, 1.5, 50);
        z = zeros(length(u), length(v));
        for i = 1:length(u)
                        for j = 1:length(v)
                        z(i,j) = mapFeature(u(i), v(j))*theta;
                        end
        end
        z = z'; %Importante transponer z antes de llamar contorno.
        %Se ecesita especificar el rango [0, 0]
        contour(u, v, z, [0, 0], 'LineWidth', 2)
end
hold off
end
```

5. Pruebas

5.1. Prueba sin Regularización

Para la primera parte, implementaremos la regresión logística sin regularizar, obteniendo los siguientes resultados:

Graficamos los datos, separando las variables de $y = 0$ y de $y = 1$.

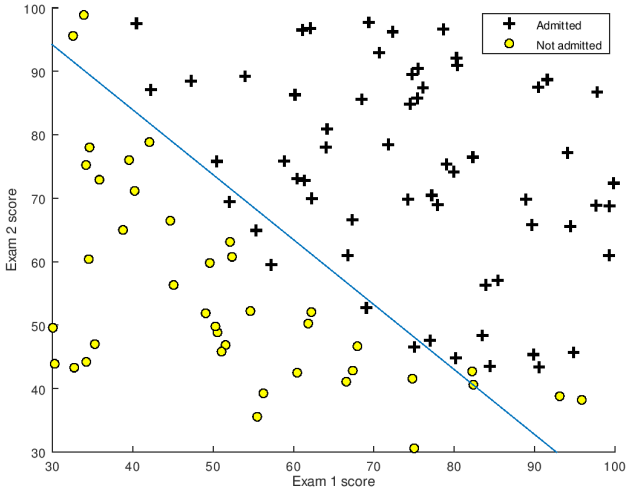


La salida en consola es la siguiente:

```
Cost at initial theta (zeros): 0.693147
Expected cost (approx): 0.693
Gradient at initial theta (zeros):
 -0.100000
 -12.009217
 -11.262842
Expected gradients (approx):
 -0.1000
 -12.0092
 -11.2628

Cost at test theta: 0.218330
Expected cost (approx): 0.218
Gradient at test theta:
```

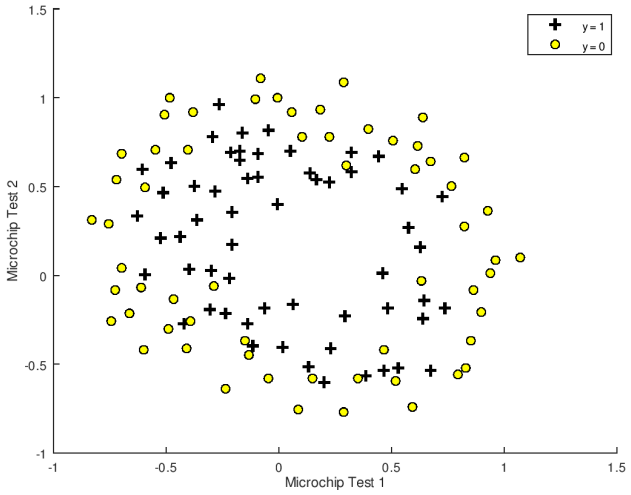
```
 0.042903
 2.566234
 2.646797
Expected gradients (approx):
 0.043
 2.566
 2.647

Program paused. Press enter to continue.
Cost at theta found by fminunc: 0.203498
Expected cost (approx): 0.203
theta:
 -25.161272
 0.206233
 0.201470
Expected theta (approx):
 -25.161
 0.206
 0.201

Program paused. Press enter to continue.
For a student with scores 45 and 85, we predict an admission probability of
Expected value: 0.775 +/- 0.002

Train Accuracy: 89.000000
Expected accuracy (approx): 89.0
```

Y la frontera de decisión que obtuvimos fue la siguiente:



5.2. Prueba con Regularización

Para nuestra prueba con regularización tenemos lo siguiente:
La información que ploteamos es la siguiente:

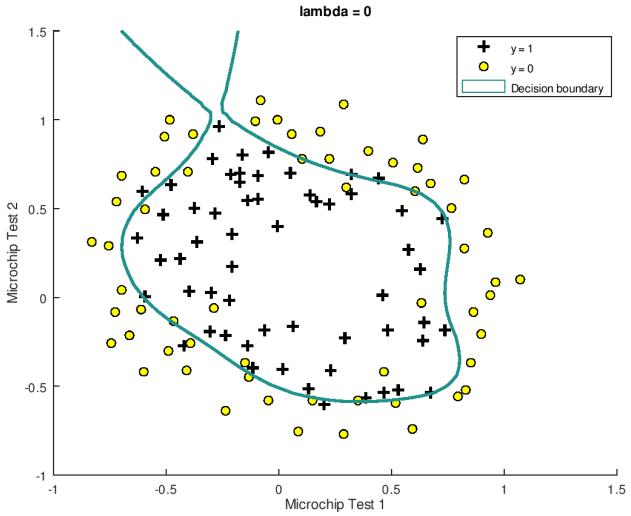


Con nuestra salida en consola es la siguiente, notese que usamos el parametro de $\lambda = 0$.

```
Cost at initial theta (zeros): 0.693147
Expected cost (approx): 0.693
Gradient at initial theta (zeros) - first five values only:
 0.008475
 0.018788
 0.000078
 0.050345
 0.011501
Expected gradients (approx) - first five values only:
 0.0085
 0.0188
 0.0001
 0.0503
 0.0115

Program paused. Press enter to continue.
```

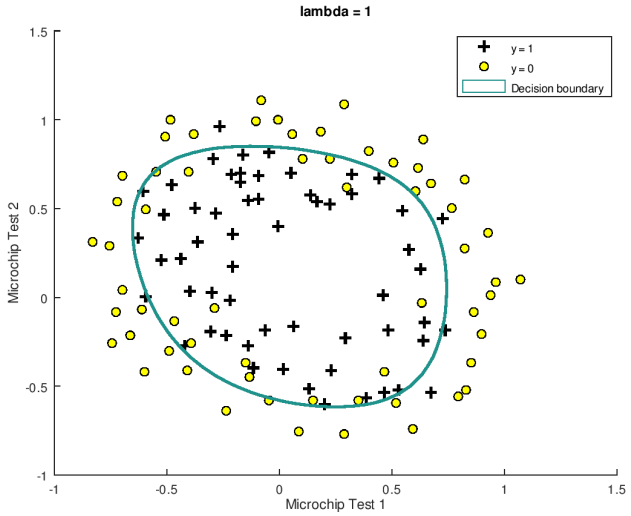
```
Cost at test theta (with lambda = 10): 3.164509
Expected cost (approx): 3.16
Gradient at test theta - first five values only:
 0.346045
 0.161352
 0.194796
 0.226863
 0.092186
Expected gradients (approx) - first five values only:
 0.3460
 0.1614
 0.1948
 0.2269
 0.0922

Program paused. Press enter to continue.
Give me the value of Lambda: 0
Train Accuracy: 86.440678
Expected accuracy (with lambda = 1): 83.1 (approx)
```

Nuestra frontera de desición con $\lambda = 0$ es la siguiente:



Si usamos el parametro de $\lambda = 0$, se produce un Overfitting. Es el efecto de sobreentrenar un algoritmo de aprendizaje con unos ciertos datos.

Si usamos el parámetro de $\lambda = 1$, tenemos lo siguiente:



Si usamos el parámetro de $\lambda = 20$, tenemos lo siguiente: