

# Segunda Tarea

## Ejercicio 1

Implementar la Función Normal y probarlo en el archivo PrimeraSemana/Datos/univariate\_reg\_data.txt de este github, y ver que los parámetros Theta obtenidos son iguales a los obtenidos por el algoritmo Gradient Descent para las siguientes funciones hipótesis:

$h(\text{Theta}) = \text{theta}_0 + x_1 * \text{theta}_1;$

$h(\text{Theta}) = \text{theta}_0 + x_1 * \text{theta}_1 + (x_1^2) * \text{theta}_2;$

$h(\text{Theta}) = \text{theta}_0 + x_1 * \text{theta}_1 + (x_1^2) * \text{theta}_2 + (x_1^3) * \text{theta}_3;$

$h(\text{Theta}) = \text{theta}_0 + x_1 * \text{theta}_1 + \sqrt{x_1} * \text{theta}_2;$

Para hallar la inversa de la matriz en Matlab/octave/python/R, utilizar la función 'pinv', la cual calcula evita errores de no invertibilidad de matrices.

Para cada hipótesis mostrar en una tabla los parámetros Theta obtenidos por el algoritmo gradient Descent en comparación con los obtenidos por la Función Normal.

## Solución

### Implementación de la función normal

```
1 function FuncionNormal(X, y, poly_degree)
2 X = composeX_norm(X, poly_degree); % llenando la matriz X con X0 y los terminos polinomiales
3 Theta = zeros(size(X, 2), 1);
4
5 Theta = pinv(X' * X) * X' * y;
6
7 disp('Ecuacion Normal -> Theta: ');
8 disp(Theta)
9
10
11 function X = composeX_norm(X, poly_degree)
12 m = size(X, 1);
13 n = size(X, 2);
14 for i_var = 1:n
15     for degree = 2:poly_degree
16         TempX = X(:,i_var) .^ degree;
17         TempX = (TempX - min(TempX)) / (max(TempX) - min(TempX)); % para normalizar.
18         X = [X, TempX];
19     end
20 end
21 X = [ones(m, 1), X];
```

### Parámetros de Theta

0	2.108	0	11	-0.4818	6.7214	22	-2.3176	4.4161
1	21.6267	0	12	-0.9781	5.4427	23	-2.2336	4.7465
2	-6.2451	0.0169	13	-1.3853	1.5184	24	-2.1299	4.4849
3	0.1801	-0.2816	14	-1.712	-2.5026	25	-2.0101	3.73
4	3.7253	2.0332	15	-1.9666	-5.0822	26	-1.877	2.6586
5	4.2065	-6.9663	16	-2.1566	-5.7538	27	-1.7336	1.4138
6	3.5224	9.905	17	-2.2894	-4.7898	28	-1.582	0.1361
7	2.5655	-0.2223	18	-2.3718	-2.8038	29	-1.4245	-1.0595
8	1.6376	-7.91	19	-2.4102	-0.4719	30	-1.2628	-2.1047
9	0.8171	-4.021	20	-2.4105	1.7244	31	-1.0984	-2.9352
10	0.1132	3.1753	21	-2.378	3.3979	32	-0.9327	-3.5201

33	-0.7669	-3.845	43	0.748	1.5676	53	1.8337	1.9059
34	-0.6019	-3.9052	44	0.8771	2.1966	54	1.9186	1.1374
35	-0.4385	-3.7518	45	1.0015	2.7251	55	1.9997	0.249
36	-0.2775	-3.3962	46	1.1212	3.1272	56	2.0771	-0.7636
37	-0.1194	-2.8694	47	1.2364	3.3931	57	2.1508	-1.888
38	0.0354	-2.2199	48	1.3469	3.5246	58	2.2211	-3.1164
39	0.1864	-1.4811	49	1.4529	3.5048	59	2.288	-4.4401
40	0.3334	-0.7065	50	1.5545	3.3292	60	2.3517	-5.8422
41	0.4761	0.0817	51	1.6518	2.9987			
42	0.6143	0.862	52	1.7448	2.5262			

## Ejercicio 2

En regresión logística, analizar el archivo PrimeraSemana/Datos/circulo\_dos\_clases.txt de este github. El cual contine 118 casos de entrenamiento, cada uno compuesto por dos variables (o características) de entrada y una variable binaria de salida {0, 1}. Probar distintas hipotesis, e encontrar aquella que mejores resultados da. (Mostrar por lo menos el grafico de 3 distintas hipotesis)

Pregunta 1: A mas variables en la función hipótesis, mayor número de iteraciones debe dar el algoritmo para converger? Considerando que en todas ellas se utiliza el mismo alfa.

Pregunta 2: Cree usted que se puede usar la función normal en la regresion logística?

## Solución

Gráfico de la función hipótesis

