

Métodos de Inteligencia Artificial

Reporte 8

IF698972

Josefina Esmeralda Arriaga Hernández

15 de marzo del 2017 Guadalajara, Jalisco

**Objetivo**

Comprender el funcionamiento de la clasificación de datos a través del perceptrón.

**Problema a resolver**

Clasificar dos series de datos que se acerca a un modelo de X grado, teniendo en cuenta que sea eficaz y preciso.

Los pasos para crear el algoritmo de perceptrón son:

1. Descargar la serie de datos
2. Realizar la regresión logística en donde se optimizan los pesos de W.
3. Realizar la simulación de cada dato en función al polinomio con los pesos óptimos de las W, en donde se verifica la exactitud, precisión y recall.
4. Graficar la frontera obtenida del modelo.

**Código desarrollado**

**Regresión logística**

%Limpieza

clear all;

close all;

clc;

%% Cargar Datos

load data2.txt;%Datos

data=data2;

G0=data(data(:,3)==0,1:2);%si columna 3=0, se agrega las primeras dos columnas

G1=data(data(:,3)==1,1:2);%si columna 3=1, se agrega las primeras dos columnas

plot(G0(:,1),G0(:,2),'bo',G1(:,1),G1(:,2),'rx')

%Se busca crear la clasificaci?n de datos

%% Regresi? Log?stica

%C?lculo de W

X=data(:,1:2);

Y=data(:,3);

m=size(X,1);%Renglones del tama?o de X

Xa=[ones(m,1) X];%lineal

Xa=[ones(m,1) X.^2]%par?bola

%Xa=[ones(m,1) X X(:,1).^2 X(:,1).\*X(:,2) X(:,2).^2];%parabola invertida

%primera columna es de unos del tama?o de m

%modelo lineal

W=zeros(size(Xa,2),1);

[J,dJdW]=func\_costo(W,Xa,Y);

options=optimset('GradObj','on','MaxIter',1000);%definir las opciones para la funcion de optimizacion

[Wopt,Jopt]=fminunc(@(W)func\_costo(W,Xa,Y),W,options)%las variables que se optimizan es W

%% Simulaci?n del Modelo Obtenido

V=Xa\*Wopt;

Yg=1./(1+exp(-V));

Yg=(Yg>=.5);%Ceros y unos

TP=sum((Y==1)&(Yg==1));%True Positive

TN=sum((Y==0)&(Yg==0));%True Negative

FP=sum((Y==0)&(Yg==1));%False Positive

FN=sum((Y==1)&(Yg==0));%False Negative

Accu=(TP+TN)/(TP+TN+FP+FN);%Exactitud

Pre=TP/(TP+FP);%Precisi?n

Rec=TP/(TP+FN);%Recall

[Accu Pre Rec]

%% Dibujar la frontera

x1=-1:0.1:1.4;

x2=-1:0.1:1.4;

[x1,x2]=meshgrid(x1,x2);% combinaciones de x1 y x2

[m,n]=size(x1);

x1temp=reshape(x1,m\*n,1);

x2temp=reshape(x2,m\*n,1);

Ytemp=[ones(m\*n,1) x1temp x2temp]\*Wopt;

Ytemp=[ones(m\*n,1) x1temp.^2 x2temp.^2]\*Wopt; %termino cuadratico

%Ytemp=[ones(m\*n,1) x1temp x2temp x1temp.^2 x1temp.\*x2temp x2temp.^2]\*Wopt;

Ytemp=reshape(Ytemp,m,n);

plot(G0(:,1),G0(:,2),'bo',G1(:,1),G1(:,2),'rx')

hold on;

contour(x1,x2,Ytemp,[0 0],'LineWidth',2)

hold off;

**Regresión logística (polinomio)**

%Limpieza

clear all;

close all;

clc;

%perceptron clasificacion supervisada

%linea adaline

%% Cargar Datos

load data2.txt;%Datos

data=data2;

G0=data(data(:,3)==0,1:2);%si columna 3=0, se agrega las primeras dos columnas

G1=data(data(:,3)==1,1:2);%si columna 3=1, se agrega las primeras dos columnas

plot(G0(:,1),G0(:,2),'bo',G1(:,1),G1(:,2),'rx')

%Se busca crear la clasificaci?n de datos

%% Regresi? Log?stica

%C?lculo de W

X=data(:,1:2);

Y=data(:,3);

m=size(X,1);%Renglones del tama?o de X

%Xa=[ones(m,1) X];%lineal

%Xa=[ones(m,1) X.^2]%parabola

%Xa=[ones(m,1) X X(:,1).^2 X(:,1).\*X(:,2) X(:,2).^2];%parabola invertido

ngrado=8;

Xa=func\_polinomio(X,ngrado);

W=zeros(size(Xa,2),1);

[J,dJdW]=func\_costo(W,Xa,Y);

options=optimset('GradObj','on','MaxIter',1000);%definir las opciones para la funcion de optimizacion

[Wopt,Jopt]=fminunc(@(W)func\_costo(W,Xa,Y),W,options);%las variables que se optimizan es W

%% Simulaci?n del Modelo Obtenido

V=Xa\*Wopt;

Yg=1./(1+exp(-V));

Yg=(Yg>=.5);%Ceros y unos

TP=sum((Y==1)&(Yg==1));%True Positive

TN=sum((Y==0)&(Yg==0));%True Negative

FP=sum((Y==0)&(Yg==1));%False Positive

FN=sum((Y==1)&(Yg==0));%False Negative

Accu=(TP+TN)/(TP+TN+FP+FN);%Exactitud

Pre=TP/(TP+FP);%Precisi?n

Rec=TP/(TP+FN);%Recall

[Accu Pre Rec]

%% Dibujar la frontera

x1=-1:0.1:1.4;

x2=-1:0.1:1.4;

[x1,x2]=meshgrid(x1,x2);% combinaciones de x1 y x2

[m,n]=size(x1);

x1temp=reshape(x1,m\*n,1);

x2temp=reshape(x2,m\*n,1);

%Ytemp=[ones(m\*n,1) x1temp x2temp]\*Wopt;%para modelo lineal

%Ytemp=[ones(m\*n,1) x1temp.^2 x2temp.^2]\*Wopt; %termino cuadratico

%Ytemp=[ones(m\*n,1) x1temp x2temp x1temp.^2 x1temp.\*x2temp x2temp.^2]\*Wopt;

Ytemp=[func\_polinomio([x1temp x2temp],ngrado)]\*Wopt;

Ytemp=reshape(Ytemp,m,n);

plot(G0(:,1),G0(:,2),'bo',G1(:,1),G1(:,2),'rx')

hold on;

contour(x1,x2,Ytemp,[0 0],'LineWidth',2)

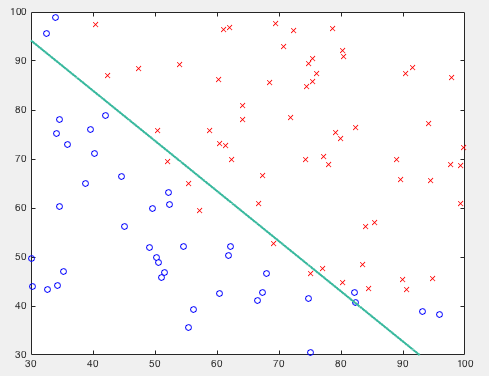
hold off;

**Gráficos**

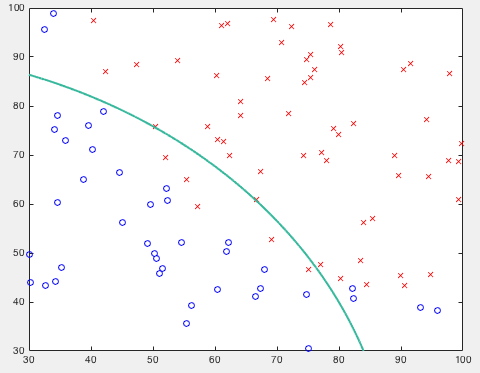
**Regresión logística**

Data 1

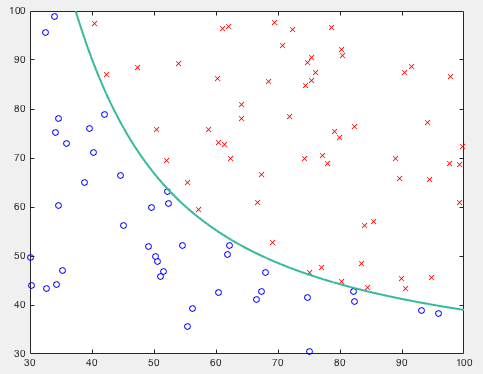
Lineal



Parábola

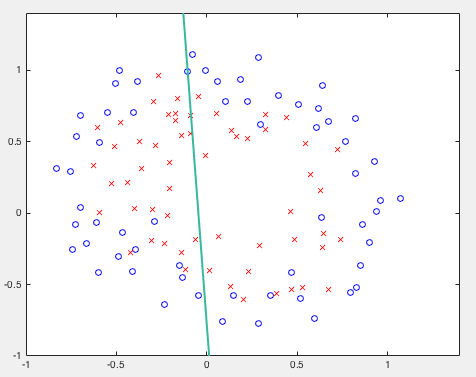


Parábola invertida

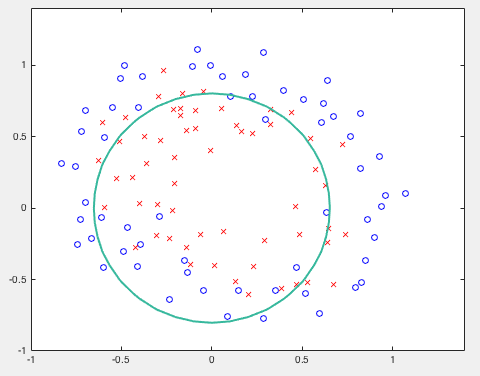


Data 2

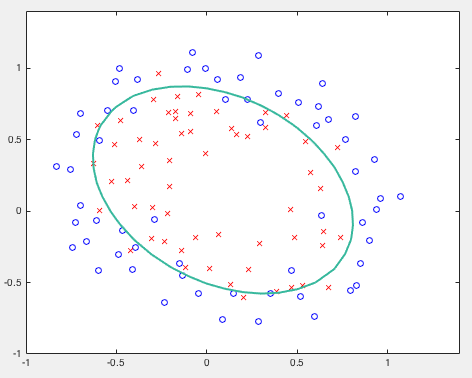
Lineal



Parábola



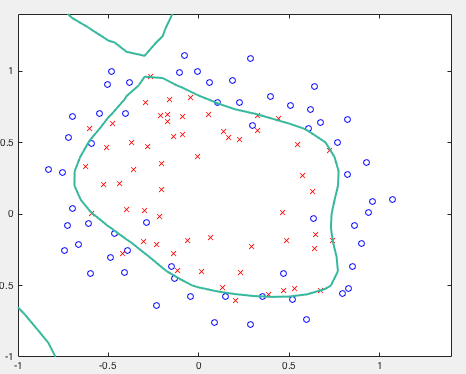
Parábola invertida



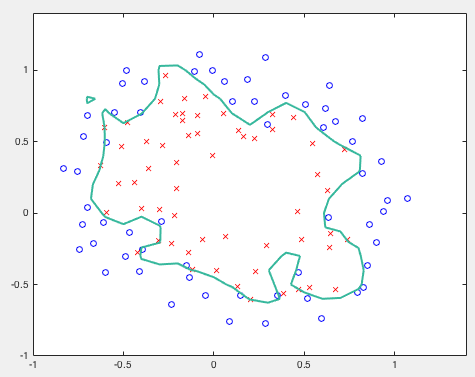
**Regresión logística (polinomio)**

**Data 2**

**Numero de grado 5**

****

**Numero de grado 8**

****

**Interpretación de gráficos**

Data 1:

A simple vista el gráfico lineal no cuadraba como modelo para la data 1, una parábola sería la mejor elección, pero tendría que ser invertida como se muestra en la tercera gráfica, haciendo que todos sus valores se encuentren “adentro” (True Positive, False Positive).

Data 2:

Al realizar lo mismo con la data 2, se encuentra con el mismo resultado que es mejor usar la tercera función para la frontera, pero sigue teniendo deficiencia, por lo tanto, se usa una función que se puede crear el polinomio al grado que se sea más útil, primero se probó con un grado 5 pero seguía siendo deficiente, por último, el gráfico muestra una frontera con una función de polinomio de grado 8, teniendo como resultado solo 4 errores en la frontera (Negative).

**Resultados**

**Regresión logística**

**Data 1**

**Lineal**

**Pesos óptimos**



**Variación óptima**

****

**Exactitud Precisión Recall**

****

**Parábola**

**Pesos óptimos**

****

**Variación óptima**

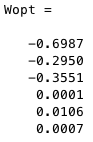
****

**Exactitud Precisión Recall**

****

**Parábola invertida**

**Pesos óptimos**

****

**Variación óptima**

****

**Exactitud Precisión Recall**

****

**Data 2**

**Lineal**

**Pesos óptimos**

****

**Variación óptima**

****

**Exactitud Precisión Recall**

****

**Parábola**

**Pesos óptimos**

****

**Variación óptima**

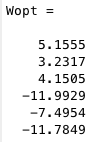
****

**Exactitud Precisión Recall**

****

**Parábola invertida**

**Pesos óptimos**

****

**Variación óptima**

****

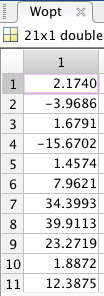
**Exactitud Precisión Recall**

****

**Regresión logística (polinomio)**

**Numero de grado 5**

**Pesos óptimos**

****

**Variación óptima**

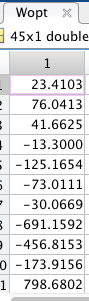
****

**Exactitud Precisión Recall**

****

**Numero de grado 7**

**Pesos óptimos**

****

**Variación óptima**

****

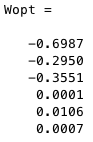
**Exactitud Precisión Recall**

****

Data 1:

Su mejor optimización fue la parábola invertida en donde su exactitud, precisión y recall da como resultado 1 y la variación es de 0.07, de manera visual también se ve que la frontera hace la separación ideal de la clasificación de datos.

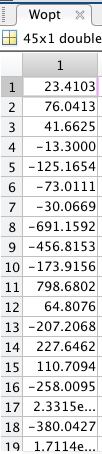
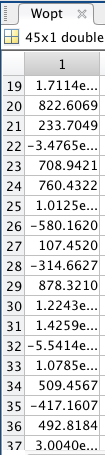
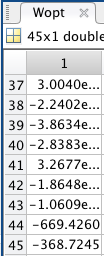
Teniendo como su W(pesos) óptimos:

****

Data 2:

Su mejor optimización es con una potencia de 7 en donde su exactitud es 0.92, su precisión es 0.90 y su recall es de 0.94, de manera visual también se observa el acercamiento óptimo de la clasificación de datos, si se usa un mayor grado se pierde la optimización.

Teniendo como su W(pesos) óptimos:



**Conclusiones**

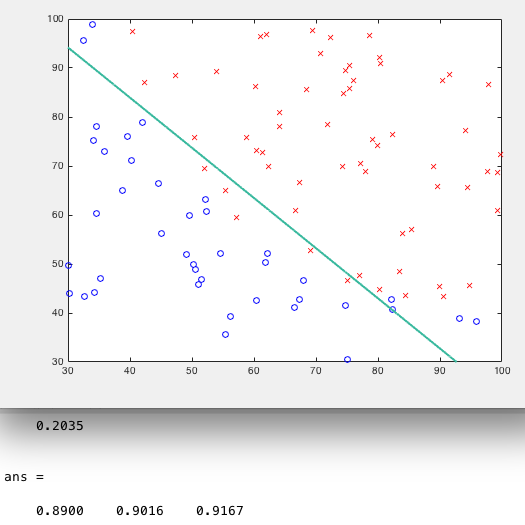
En conclusión, considero que la optimización de datos por el método de adaline o por el método de perceptrón tienen ambos sus pros y contras, pero lo que en verdad hace la diferencia es analizar el grafico de los datos y saber la forma en la que se debería tener la frontera para saber el modelo adecuado que encuentre los pesos óptimos de W. De igual manera tener los parámetros de exactitud precisión y recall son de gran ayuda para saber si el modelo tiene el mejor acercamiento.

**Funcionamiento**

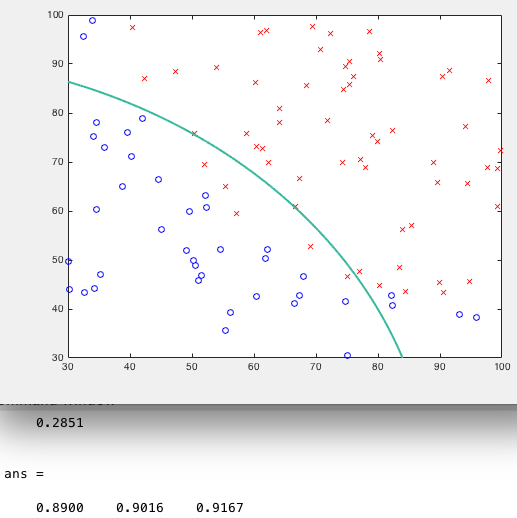
**Regresión logística**

**Data 1**

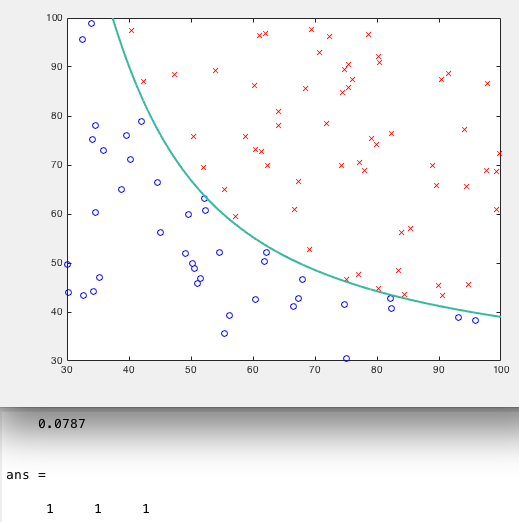
**Lineal**

****

**Parábola**

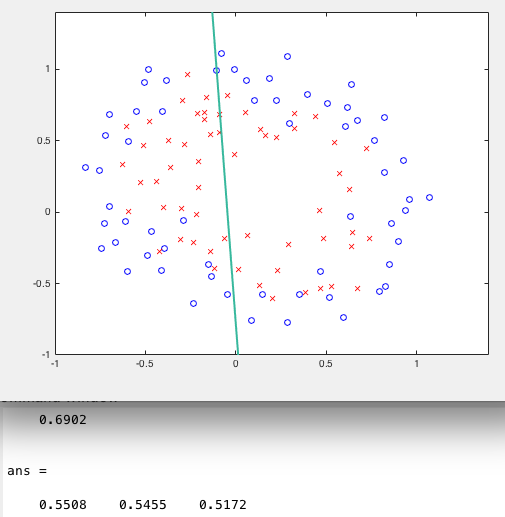
****

**Parábola invertida**

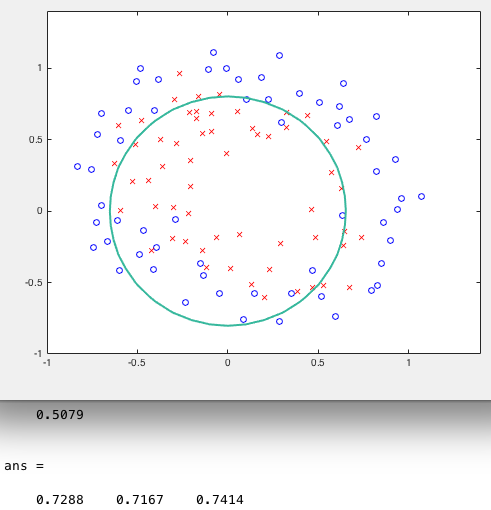
****

**Data 2**

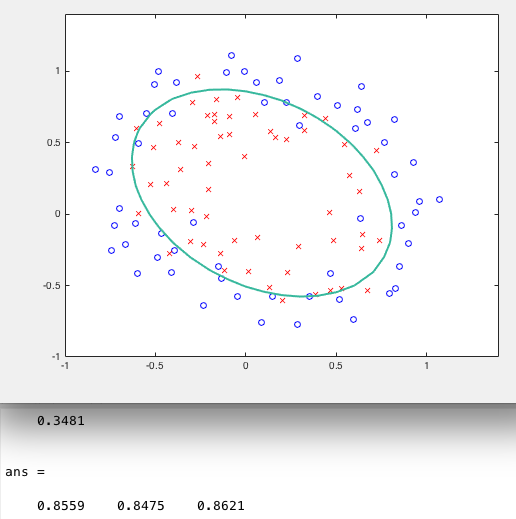
**Lineal**



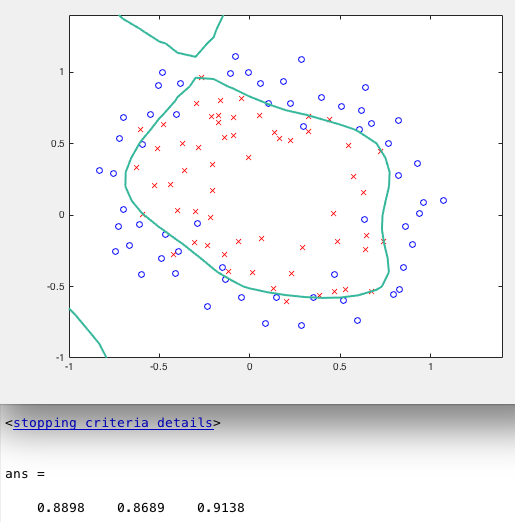
**Parábola**

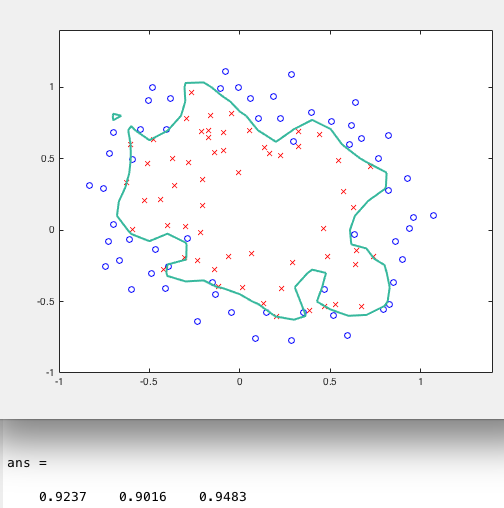


**Parábola invertida**



**Regresión logística (polinomio)**

****

****