ALGORITHM FOR LOGICAL REGRESSION (BY USING GRADIENT DESCENT)

// Firstly we have to import data set from file to Matlab.

df=importdata('Data1.csv');

x1=df(:,1);

x2=df(:,2);

y=df(:,3);

sum1=0;

sum2=0;

//Classes in the file was given mixed way.So we should make them seperate.

for i=1:100

if y(i)==1

sum1=sum1+1;

plot(x1(i),x2(i),'kx','Color','r','MarkerSize',4)

hold on

class1(sum1,1)=(x1(i));

class1(sum1,2)=(x2(i));

else

sum2=sum2+1;

plot(x1(i),x2(i),'ko','Color','b','MarkerSize',4)

hold on

classminus1(sum2,1)=(x1(i));

classminus1(sum2,2)=(x2(i));

end

end

X0=ones((sum2+sum1),1);

X1=[class1(:,1);classminus1(:,1)];

X2=[class1(:,2);classminus1(:,2)];

X\_general\_unnormalize=[X0 X1 X2];

X\_general=[ X0 mat2gray(X1) mat2gray(X2)];

Ygen=[zeros(size(class1(:,1))) ; ones(size(classminus1(:,1)))];

training\_number=length(X0);

theta=randi([20,200],3,1); //initial theta values

error\_func=-1./(training\_number).\*(Ygen.\*log(1./(1+exp(-(teta(1,1).\*X\_general(:,1)+theta(2,1).\*X\_general(:,2)+theta(3,1).\*X\_general(:,3)))))+(1-Ygen).\*log(1-1./(1+exp(-(theta(1,1).\*X\_general(:,1)+theta(2,1).\*X\_general(:,2)+theta(3,1).\*X\_general(:,3)))))); //cost function

learning\_rate=16;

epsilon=1e-10\*ones(length(X0),1);

iteration=0;

maxiteration=800000;

while sum(abs(error\_func(:,1))>epsilon)~=0 && (maxiteration>iteration)

iteration=iteration+1;

for i=1:1:training\_number

temp1=theta(1,1)-learning\_rate\*(1/(1+exp(-(theta(1,1)\*X\_general(i,1)+theta(2,1)\*X\_general(i,2)+theta(3,1)\*X\_general(i,3))))-Ygen(i,1))\*X\_general(i,1);

temp2=theta(2,1)-learning\_rate\*(1/(1+exp(-(theta(1,1)\*X\_general(i,1)+theta(2,1)\*X\_general(i,2)+theta(3,1)\*X\_general(i,3))))-Ygen(i,1))\*X\_general(i,2);

temp3=theta(3,1)-learning\_rate\*(1/(1+exp(-(theta(1,1)\*X\_general(i,1)+theta(2,1)\*X\_general(i,2)+teta(3,1)\*X\_general(i,3))))-Ygen(i,1))\*X\_general(i,3);

theta(1,1)=temp1;

theta(2,1)=temp2;

theta(3,1)=temp3;

error\_func(i,1)=-1/(training\_number)\*(Ygen(i,1)\*log(1/(1+exp(-(theta(1,1)\*X\_general(i,1)+theta(2,1)\*X\_general(i,2)+theta(3,1)\*X\_general(i,3)))))+(1-Ygen(i,1))\*log(1-1/(1+exp(-(theta(1,1)\*X\_general(i,1)+theta(2,1)\*X\_general(i,2)+theta(3,1)\*X\_general(i,3))))));

end

end

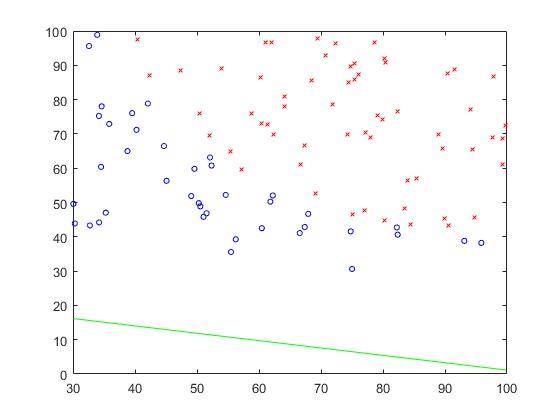
h\_theta\_x= theta(1,1).\*X\_general(:,1)+theta(2,1).\*X\_general(:,2)+(-(theta(1,1)/theta(3,1)))-((theta(2,1)/theta(3,1)).\*X\_general(:,2));

plot(X1,h\_theta\_x,'Color','g');

hold on;

DISCUSSIONS

I selected learning rate as 1 but iteration took so much time to train the algorithm.I restricted the iteration with a high value.As a result training the algorithm was so bad.Like this way;



After so many trials I selected training algoritm as 16.And at that time classification border seperated classes from each other. That is, Class 1 and Class -1. Unfortunately there is error.Prediction is not excellent.

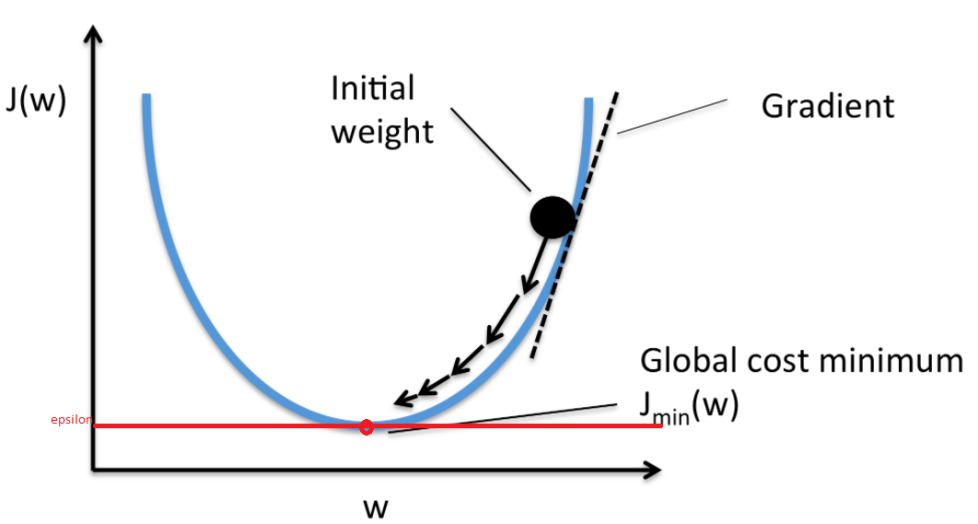


H(theta\_x)=theta\_1+theta\_2\*x1+theta\_3\*x2 🡪 this equation is a model for logistic regression.And after iterations , theta values was found as;

theta\_1 = 94.1628

theta\_2 = -78.8055

theta\_3 = -75.8782



When cost function (j or error) intersect with epsilon value.Error will be minimized.

(w=theta)