רגרסיה לוגיסטית

תרגיל כיתה – הדרכה לתרגיל בית

בתרגיל זה נעשה שימוש באלגוריתם ה- Gradient Descent למימוש רגרסיה לוגיסטית לפתרון בעיית סווג.

נניח כי קבוצת נתוני האימון מייצגת תוצאות מבחני קבלה של 80 סטודנטים, מהם 40 שהתקבלו למכללה ו- 40 שלא התקבלו. כל דגימת אימון i (סטודנט) מיוצגת על-ידי $\left(X^{(i)}, y^{(i)}\right)$, כאשר המשתנה הראשון $\left(X^{(i)}\right)$ הוא וקטור המכיל את ציוני שני המבחנים, והמשתנה השני $\left(y^{(i)}\right)$ הוא סקלר – תגית המסמנת אם הסטודנט התקבל למכללה (1) או לא התקבל (0).

בנו מערכת מסווג בינארי באמצעות רגרסיה לוגיסטית המשערכת את סיכויי סטודנטים להתקבל למכללה על סמך תוצאות שני מבחנים.

נתוני האימון נמצאים באתר הקורס במודל במחיצה Materials for Ex. 2 - Logistic Regression (X.dat ו- y.dat).

א. טענו את הנתונים לחלון העבודה באמצעות הפקודות הבאות:

```
%% College admittance decision using logistic regression
clc
%load data
load X.dat
load y.dat
% X is the feature matrix (each row represents one student grades)
% y - the corresponding label (1- admitted, 0 - not admitted)
pos_index=find(y==1);
neg_index=find(y==0);
```

ב. איתחול ונירמול הנתונים:

```
%% Initialization
m=length(y); % number of training examples
alpha=0.01; % here we should use values of alpha of the following
% orders:
% alpha = 1e-4, or 3.33e-4 or 1e-3, or 3.33e-3, 1e-2, 3.33e-2 etc.
X1=[ones(m,1) X ]; % adding a first column of ones
% normalizing the input values
for j=2:size(X1,2)
    X2(:,j)=(X1(:,j)-mean(X1(:,j)))/std(X1(:,j));
end
theta=zeros(size(X1,2),1);
```

ג. הפעלת אלגוריתם gradient descent

```
%% Applying gradient descent
num_iters=2500;
[theta,J]=gd(X2,y,theta,alpha,num_iters);
```

הפונקציה gd מבצעת את אלגוריתם ה- gradient descent.

עליכם לכתוב פונקציה בה העדכון יתבצע עבור כל דוגמאות האימון (Batch), ועבור כל התכונות באופן מטריצי.

פונקציית המחיר אותה יש למזער היא:

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

או באופן מפורט:

$$\begin{aligned}J(\theta)=-\frac{1}{m}(\ & y^{(1)}\log\left(h_\theta(x^{(1)})\right)+\left(1-y^{(1)}\right)\log\left(1-h_\theta(x^{(1)})\right)+\\ & y^{(2)}\log\left(h_\theta(x^{(2)})\right)+\left(1-y^{(2)}\right)\log\left(1-h_\theta(x^{(2)})\right)+\dots\\ & y^{(m)}\log\left(h_\theta(x^{(m)})\right)+\left(1-y^{(m)}\right)\log\left(1-h_\theta(x^{(m)})\right)\ )\end{aligned}$$

נתבונן בביטוי הראשון בכל אחד מהמחוברים :

המשתנה $y^{(i)}$ הוא סקלר, וכן $h_\theta(x^{(i)})$. לכן החישוב של $J(\theta)$ באופן מטריצי (וקטורי) הוא כפל הוקטור y המכיל את כל התגיות בוקטור $h_\theta(x)$, המכיל את פונקציית ההיפותיזה עבור כל דוגמאות האימון :

```
y'*log(h_theta)
```

באופן זה נבצע את חישוב פונקציית המחיר לכל איטרציה באופן הבא :

```
m = length(y); % number of training examples
J = 0; % Initial cost
theta=theta(:);
y=y(:);
grad_J = zeros(size(theta));

% Computing the hypothesis function for theta using sigmoid function
h_theta=sigmoid(X*theta);

J=1/m*(-y'*log(h_theta)-(1-y')*log(1-h_theta)); % Cost function

grad_J=1/m*X'*(h_theta-y); % Gradient
```

כתבו פונקציה שתבצע את חישוב $J(\theta)$ לכל איטרציה. הפונקציה תקבל בכניסה את ערכי theta, X, ו- y, ותוציא את פונקציית המחיר ואת ערך הגרדיאנט לכל איטרציה.

הפונקציה gd המתוארת בהמשך, תשתמש בפונקציה זו לחישוב הפרמטרים θ.

```
function [theta, J,grad] = gd(X, y, theta, alpha, num_iters)
% gd Performs gradient descent to obtain theta
%   theta = gd(X, y, theta, alpha, num_iters) updates theta by
%   using num_iters gradient steps with learning rate alpha
% Input arguments:
% X a matrix whose rows are the input feature vectors, where the
first
% column is 1's.
% y - the corresponding labels of the rows in X ( 0 or 1).
% theta - initial theta
% alpha - the learning rate
% num_iters - maximum number of iterations
% Output arguments
% theta - the parameters for logistic regression 1./(1+exp(-X*theta))
% J - cost function J(theta)
% grad - the gradient of the cost function
% Usage: [theta, J,grad] = gd(X, y, theta, alpha, num_iters)
%
```

ד. ציור הנתונים.

1. בשלב זה נצייר את נקודות נתוני האימון:

```
figure(1)
plot(X2(pos_index,2), X2(pos_index,3),'ro', X2(neg_index,2),…
X2(neg_index,3),'xg','MarkerSize', 5, 'lineWidth',2.5)
xlabel('exam 1'), ylabel('exam 2'),
title('training data for admittance classifier')
figure(1)
pause(0.1)
hold on
```

2. עתה נרצה לבחון את משטח ההחלטה. במקרה זה וקטור התכונות מכיל רק שתי תכונות (הציונים של שני המבחנים) ולכן משטח ההחלטה הוא עקום על המישור. מאחר ו-

$\theta^T X = \theta_0 + \theta_1 x_1 + \theta_2 x_2$, משטח ההחלטה הוא ישר. ההחלטה עבור כל סטודנט או דגימה מקבוצת האימון היא "התקבל" (1) אם:

$$g\left(\theta^T X\right) = g\left(\theta_0 + \theta_1 x_1 + \theta_2 x_2\right) = \frac{1}{1 + e^{-\left(\theta_0 + \theta_1 x_1 + \theta_2 x_2\right)}} > 0.5$$

תנאי זה מתקיים כאשר $\theta^T X = \theta_0 + \theta_1 x_1 + \theta_2 x_2 > 0$.

באופן דומה ההחלטה עבור כל סטודנט או דגימה מקבוצת האימון היא "לא התקבל"(0) אם:

$$g\left(\theta^T X\right) = g\left(\theta_0 + \theta_1 x_1 + \theta_2 x_2\right) = \frac{1}{1 + e^{-\left(\theta_0 + \theta_1 x_1 + \theta_2 x_2\right)}} < 0.5$$

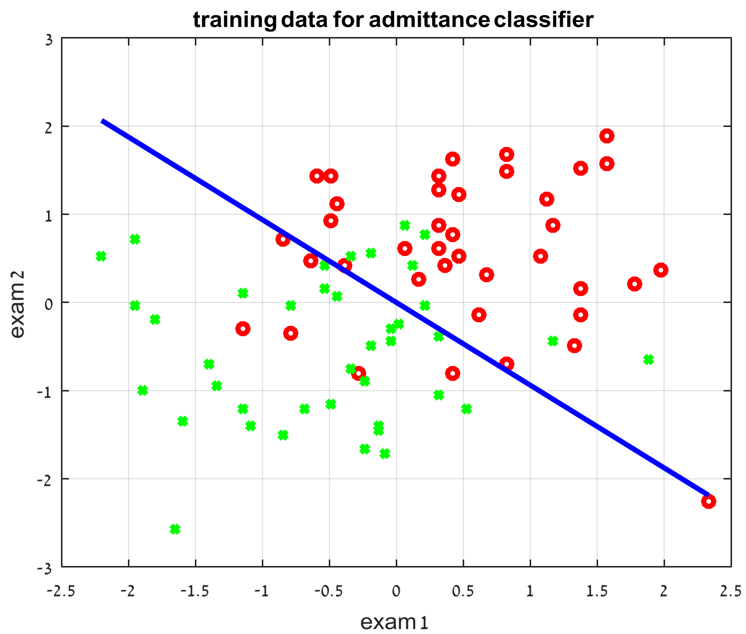
תנאי זה מתקיים כאשר $\theta^T X = \theta_0 + \theta_1 x_1 + \theta_2 x_2 < 0$.

משוואת הישר המפריד היא אם כן: $\theta_0 + \theta_1 x_1 + \theta_2 x_2 = 0$, ולכן:

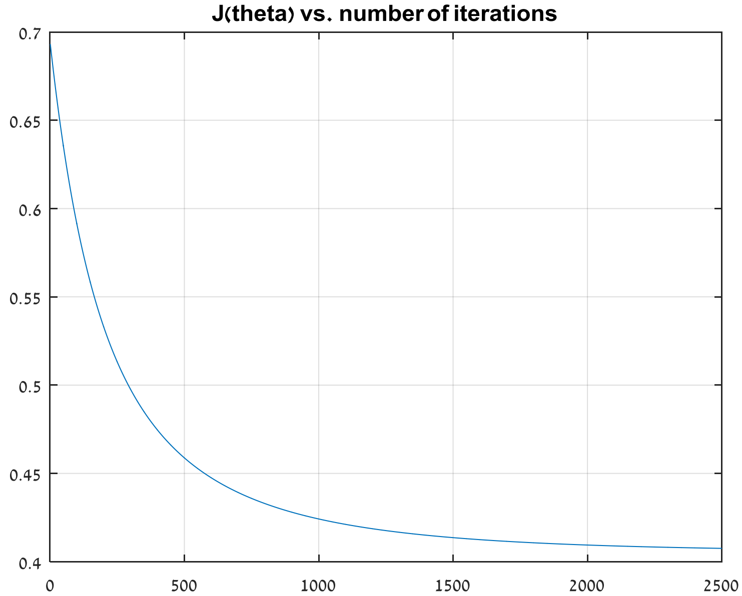
$$x_2 = \frac{-\theta_0 - \theta_1 x_1}{\theta_2}.$$

נצייר את משטח ההפרדה על-ידי שימוש בפקודת line:

```
% please see help line
line([min(X2(:,2)),max(X2(:,2))],...
     [(-theta(1)-theta(2)* min(X2(:,2)))/theta(3),...
     (-theta(1)-theta(2)* max(X2(:,2)))/theta(3)],...
     'LineWidth', 2.5, 'Color', 'b')
grid
hold off
```



3. נבחן האם אלגוריתם ה- gd מתכנס וכן האם הפרמטרים (alpha, num_iters) מתאימים על-ידי ציור פונקציית המחיר J כתלות במספר האיטרציות.



ה. עתה נבחן האם הסטודנט יתקבל באופן אוטומטי (כלומר על-פי החלטת המסווג) אם ציוני המבחנים שלו 35 במבחן הראשון ו- 85 במבחן השני.
מהי ההסתברות של סטודנט להתקבל למכללה? (פתרון: ההסתברות היא 0.6871).

```
student1_grades=[1 25 85];
for j=2:size(X2,2)
    student1_grades(:,j)=(student1_grades(:,j)-
mean(X1(:,j)))/std(X1(:,j));
end

p_admitted=sigmoid(student1_grades*theta)
```