coursera

Discussion Forums

Week 3

SUBFORUMS

All

Assignment: Logistic Regression

← Week 3

Regarding how plotDecisionBoundary() works

Tom Mosher Mentor · a year ago

This post derives how the plotDecisionBoundary() function works.

For logistic regression, h = sigmoid(X * theta). This describes the relationship between X, theta, and h.

We know theta (from gradient descent).

We know h - by definition, the decision boundary is the locus of points where h = 0.5, or equivalently (X * theta) = 0, since the sigmoid(0) is 0.5.

Now we can write out the equation for the case where we have two features and a bias unit, and we write X as $[x_0 x_1 x_2]$ and theta as $[\theta_0 \theta_1 \theta_2]$

$$0 = x_0\theta_0 + x_1\theta_1 + x_2\theta_2$$

x_0 is the bias unit, it is hard-coded to 1.

$$0 = \theta_0 + x_1\theta_1 + x_2\theta_2$$

Solve for x_2

$$x_2 = -(\theta_0 + x_1\theta_1)/\theta_2$$

Now, to draw a line, you need two points. So pick two values for x_1 - anything near the minimum and maximum of the training set will serve. Compute the corresponding values for x_2, and plot the $(x_1 x_2)$ pairs on the horizontal and vertical axes, then draw a line through them.

This line represents the decision boundary.

This is exactly what the plotDecisionBoundary() function does. x_2 is the variable "plot_y", and x_1 is the variable "plot_x".

==========

keywords: tutorial plotDecisionBoundary()

33 Upvote · Follow 16 · Reply to Tom Mosher

This thread is closed. You cannot add any more responses.

Earliest **Top** **Most Recent**

AZ

Amy Zheng · 2 months ago · Edited

Why do we solve for x2 specifically and not x1? x2=−(θ0+x1θ1)/θ2. Can we do it to x1?

0 Upvote · Hide 2 Replies

F

Falconwing17 · 2 months ago

I would suppose you could. You could pick two values of x2 and then take the corresponding values of x1 instead.

0 Upvote

Tamas Simahazi · 2 months ago

I think it's explained by the last sentence in the comment:

x2 is the variable "plot_y", and x1 is the variable "plot_x"

We started with picking two values for "plot_x", and then we are trying to find out the formula for "plot_y", that's why we are solving the equation for x2.

If we started with picking two values for "plot_y", and afterwards try to find the formula for "plot_x", then this time we should solve the same equation for x1. It would give the same result in the end.

0 Upvote

Giovanni De Cillis · 7 months ago

Hi Tom,

Thank you for this and other explanations!

I am trying to plot, on the same frame, different decision boundaries for different lambda values .

This is the idea I am following:

I write a loop for fminunc and I add lamdba as variable in plotDecisionBoundary function.

Lambda is a vector now.

I am using hold on.

I can produce the graphs but they are not in the same frame.

Can you help me?

Thank you!

0 Upvote · Hide 5 Replies

Tom Mosher Mentor · 7 months ago

Search the Mathworks web page and see if you can find a tutorial on plotting functions.

0 Upvote

Giovanni De Cillis · 7 months ago

I checked Tom...

My question comes from the observation that we plot using another function, we don't use plot in the main document, so I wasn't sure about

the use of hold on. Thank you anyway.

0 Upvote

Tom Mosher Mentor · 7 months ago · Edited

Some of the plotting in these exercises is a bit complicated. They create a plot in one function, then use "hold on" in another function to add data or the decision boundary to it. It gets rather confusing.

The exercise code would be better structured if the plots were all built in one function, rather than being spread out.

2 Upvote

Giovanni De Cillis · 7 months ago

I see,

how can you do this? I mean, how can you make sure that plots are all built in as a module?

Thank you!

0 Upvote

Tom Mosher Mentor · 7 months ago

They would have needed to built a function that takes all of the data to be plotted (the training set, and the theta values that define the decision boundary), and the axis labels and data legends) and create the plot figure all in one function.

0 Upvote

TH

Tri Han · 8 months ago

awesome explanation. thank you, Tom

0 Upvote · Reply

Md. Enzam Hossain · 8 months ago

Thanks for the explanation.

I have a question on the else part.

I can't figure out why we need to transpose z before plotting.

Can you please explain the logic behind this?

0 Upvote · Hide 1 Reply

Tom Mosher Mentor · 8 months ago

Use the commands "help contour" and "help contourc" to discover the reason.

0 Upvote

Parnika · 8 months ago

Helpful explanation but in the else's part of this function, I need to know that in linspace why base and limit is -1 and 1.5 respectively?Also in mapFeature why the degree is 6?

Kindly help me out here.

0 Upvote · Hide 3 Replies

Tom Mosher Mentor · 8 months ago · Edited

Those values cover the range of X values for this exercise.

In my copy of the function, I've replaced those lines with this, so it works for any set of data:

```
u = linspace(min(X(:,2)), max(X(:,2)), 50);
v = linspace(min(X(:,3)), max(X(:,2)), 50);
```

2 Upvote

Tom Mosher Mentor · 8 months ago

And the degree was set to 6 by the authors of this exercise because it worked well enough for the lesson they were teaching. You can experiment with different values.

0 Upvote

Parnika · 8 months ago

Helpful enough!

0 Upvote

DK

David King · 9 months ago

Thanks for the explanation. This seems so obvious now! ;)

0 Upvote · Reply

MS

Murtuza Shareef · a year ago

Thanks for explaining this!! Appreciate much.

1 Upvote · Reply

Kevin Zakka · a year ago

Thanks for this, helped a lot!

0 Upvote · Reply

AS

Anand Sankar · a year ago

Hi Tom,

Thanks a lot for that explanation.

Would it be possible for you to explain the second part of the same function pertaining to the non-linear case(N>3) ?

1 Upvote · Hide 5 Replies

Tom Mosher Mentor · a year ago

The code creates a grid of feature values for the horizontal and vertical axes. It adds the quadratic terms and computes the linear hypothesis value, and creates a contour plot of the surface where the value is 0. This is equivalent to the logistic hypothesis value 0.5.

1 Upvote

xiang zhou · a year ago

Hi Tom,

I am still a bit of confused that

for i = 1:length(u)

for j = 1:length(v)

for j = 1:length(v)

z(i,j) = mapFeature(u(i), v(j))*theta;

end

end

is a 50*50 loop while

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

is quite different number of loop?

It seems like it adds the quadratic terms in mapfeature for only X1 and X2.

what if I have X1...Xn features? is it still possible to visualise the decision boundary?

What does " creates a grid of feature values for the horizontal and vertical axes." mean? the only thing that I am clear is we need to figure out the boundary when X*theta = 0.

sorry that I throw so many questions at you!

I am a bit confused about the second part still.

thanks in advance!

erik

0 Upvote

Tom Mosher Mentor · a year ago

Sorry, I don't have anything to add.

0 Upvote

xiang zhou · a year ago

Thanks anyway Tom, I will figure out the code part!

I can understand 2D or 3D graph, my question is is it possible to plot a graph that is more than 3 dimensions, say I have variables :x1,x2,x3,x4,x5,x6,x7 or even xn?

0 Upvote

Tom Mosher Mentor · a year ago

A 3-D plot plus color would give you four features. Beyond that, no.

2 Upvote