Discussion Forums

Week 2

SUBFORUMS

All

Assignment: Linear Regression

Week 2

ex1 tutorial for featureNormalize()

M

Tom Mosher Mentor Week 2 · 3 years ago · Edited

This tutorial discusses how to implement the featureNormalize() function using vectorization.

There are a couple of methods to accomplish this. The method here is one I use that doesn't rely on automatic broadcasting or the bsxfun() or repmat() functions.

- You can use the mean() and sigma() functions to get the mean and std deviation for each column of X. These are returned as row vectors (1 x n)
- Now you want to apply those values to each element in every row of the X matrix. One way to do this is to duplicate these vectors for each row in X, so they're the same size.

One method to do this is to create a column vector of all-ones - size (m x 1) - and multiply it by the mu or sigma row vector (1 x n). Dimensionally, (m x 1) * (1 x n) gives you a (m x n) matrix, and every row of the resulting matrix will be identical.

- Now that X, mu, and sigma are all the same size, you can use element-wise operators to compute X_normalized.

Try these commands in your workspace:

```
X = [1 2 3; 4 5 6]        % creates a test matrix
mu = mean(X)              % returns a row vector
sigma = std(X)            % returns a row vector
m = size(X, 1)            % returns the number of rows in X
mu_matrix = ones(m, 1) * mu
sigma_matrix = ones(m, 1) * sigma
```

Now you can subtract the mu matrix from X, and divide element-wise by the sigma matrix, and arrive at X_normalized.

You can do this even easier if you're using a Matlab or Octave version that supports automatic broadcasting - then you can skip the "multiply by a column of 1's" part.

You can also use the bsxfun() or repmat() functions. Be advised the bsxfun() has a non-obvious syntax that I can never remember, and repmat() runs rather slowly.

54 Upvotes   Reply   Follow this discussion

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

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

AR

Alejandro Rojas · 2 years ago

Tom, great tutorials, really apretiate it! Cheers

4 Upvotes   Reply

M

Tom Mosher Mentor · 2 years ago

You misunderstand how the mean() function works. Use the "help mean" command in the console for the details.

The second parameter says what dimension to use, not which column.

'1' means compute the mean of the rows.

'2' means compute the mean of the columns.

4 Upvotes   Reply

C

Célien · 2 years ago · Edited

Hi Tom,

Thanks a lot for your tutorials, they are very helpful.

I try to use the mean function without much success.

To compute the mean of the second column ox X, I just do

```
mean(X,2)
```

And it works well (I got a 1 * n vectors returned).

However, if I want to do the same for the first column:

```
mean(X,1)
```

The result is a 2 * n matrix. Why?

0 Upvotes Reply

GK

Gowthami Kankanalapalli · 2 years ago

Hi Tom,

I have followed the tutorial as shown at the top for the below example and understood the logic.

X = [1 2 3; 4 5 6] % creates a test matrix

mu = mean(X) % returns a row vector

sigma = std(X) % returns a row vector

m = size(X, 1) % returns the number of rows in X

mu_matrix = ones(m, 1) * mu

sigma_matrix = ones(m, 1) * sigma

However, for the assignment, i am confused why the mu and sigma values are initialized to a matrix of 1xno.of columns in X size with all values zero.

How do I define the value of m which is the factor I need to compute mu_matrix and sigma_matrix?

Thanks!

1 Upvote Reply

Rohan Marwaha · 2 years ago

Hi Tom,

I followed the tutorials, But I'm still getting an error.

Not enough input arguments.

Error in featureNormalize (line 9)

X_norm = X;

How can I rectify this?

0 Upvotes Hide 3 Replies

Tom Mosher Mentor · 2 years ago

Please read the "FAQ for Week 2 and programming exercise 1" in the Week 2 forum area.

0 Upvotes

Rohan Marwaha · 2 years ago

What I mean is, how can

X_norm = X;

create an error. This part of the code isn't even written by the student and we aren't supposed to edit it.

1 Upvote

Tom Mosher Mentor · 2 years ago

Did you read Q10 of the FAQ?

Please post a screen caputure of the command you entered that generated this error.

0 Upvotes

EI Edwin ismail · 2 years ago

I still can't find the normalized

0 Upvotes Reply

Pawel Tyszka · 2 years ago

Thanks for the tutorial Tom. I did the same as you suggested so I multiplied mu by ones(47,1) and got submission error. Only after changing it to ones(size(X,1), 1) my submission was accepted. Why was I getting the error when hard-coding 47 if it's the correct dimension?

0 Upvotes Hide 2 Replies

Tom Mosher Mentor · 2 years ago

47 is only the correct value if every data set in the entire world had exactly 47 training examples.

Your code must work with any size of data set. The submit grader's test case is different than the one in the exercise script.

0 Upvotes

Pawel Tyszka · 2 years ago

That's very useful! Thanks!

0 Upvotes

Rayed Bin Wahed · 2 years ago

Why is it necessary to compute mu_matrix and sigma_matrix?? Works just fine with mu and sigma.

1 Upvote Hide 1 Reply

Tom Mosher Mentor · 2 years ago

If you are just using mu and sigma, they are both vectors. Mathematically, you cannot use a vector to modify a matrix.

Octave allows you to do this by an evil trick called "automatic broadcasting". But relying on it is risky, and your code will not be portable to MATLAB.

2 Upvotes

HJ Haseeb Javed · 2 years ago

Hi Tom,

I've been playing around with Alpha and num_iterations a lot but keep on getting $293665.543943 when using Gradient Descent.

My price answer when using the Normal Equation is correct: $293081.464335.

Any advice? It's really frustrating to be this close. Do you think my featureNormalize(X) function is not implemented well?

I've used Alphas of 0.01, 0.03, 0.1, 0.3, etc. and num_iterations of 1000, 5000, etc. and all of their combinations together but I can't move the needle from $293665.xxxx. Only the tenth or hundredth cent values change.

Thank you,

Haseeb

0 Upvotes Hide 3 Replies

HJ Haseeb Javed · 2 years ago

FYI - I calculated the normalized value for the house size of 1650 and it came out to be -0.44117 if done manually on a calculator but -0.43584 when using featureNormalize().

Thanks!

0 Upvotes

HJ Haseeb Javed · 2 years ago

My bad. I figured it out. I was incorrectly adding the test data [1650 3] to the matrix X and then normalizing all of it. I was doing this so that I could easily extract the normalized test data once featureNormalize() was complete.

However this was giving me incorrect normalized data for [1650 3] because this additional row vector skewed the mean and std.

The normalization of X happens as part of ex1_multi.m to compute theta.

And then I'm normalizing the test data [1650 3] by performing the normalization in manual steps and then calculating price.

Is this manual way the correct and most efficient way to calculate the price? Are we supposed to normalize manually first and then apply the theta * [normalized test data] calculation when trying to determine price?

Feels like there should be a quicker way to do so.

0 Upvotes

Tom Mosher Mentor · 2 years ago

Yes, that is the correct method.

1 Upvote

AS Aniket Prakash Srivastava · 2 years ago

I quote the text from exercise 1 , page 11: "You will do this for all the features and your code should work with datasets of all sizes (any number of features / examples)."

Now for this last bit, I am trying to make the code a generic one feeding in "1: size(data,2)-1" to initiate X and size(data,2) to initiate y. Is this a correct working (my code works fine) from a coding point of view.

0 Upvotes Hide 1 Reply

Tom Mosher Mentor · 2 years ago

You don't have to do that. The ex1.m script does that for you.

Your code just operates on the values that are passed to it.

0 Upvotes

Tom Mosher Mentor · 2 years ago

Check you're using the right variable names. Inside featureNormalize(), you're operating on X and saving it in the local variable X_norm. X_norm is what gets returned.

In the template code, X_norm is initialized as a copy of X. So if you still have un-normalized values after calling featureNormalize(), it could be that your code isn't updating X_norm inside the function.

1 Upvote Hide 2 Replies

JC João Paulo Caldeira · 2 years ago

Hi Tom.

After normalization shouldn't the first column of X still be equal to 1?

I only get "Nice work" after normalization returns NaN in first column.

0 Upvotes

Tom Mosher Mentor · 2 years ago

Normalize first, then add the bias unit.

0 Upvotes

Nicolas Kokel · 2 years ago · Edited

Nevertheless I kept stuck.

I have typed all lines of codes to normalize X:

- lines 38 to 40 from ex1_multi.m
- 'my own lines of codes' to modify featureNormalize.m
- line 55 from ex1_multi.m

and the result is the correct normalized 47x3 dimensional X vector with the first column made of ones and normalized surface and room number values.

When however I type the EXACT SAME 'my own lines of codes' in featureNormalize.m file, save it, clear and and redo command >> ex1_multi in Octave, and after that type X at the prompt, I get a 47x3 dimensional X vector with non-normalized values for surface and room number in column 2 and 3.

'my own lines of codes' are:

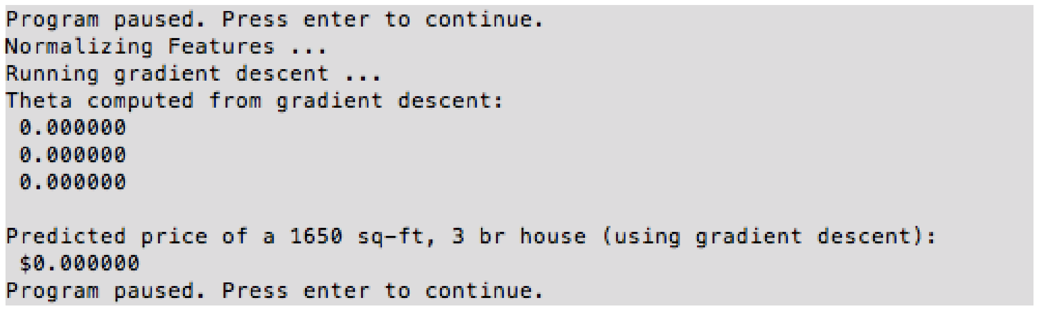
- calculating mu
- calculating sigma
- defining m
- building the mu matrix by making ones(m,1) times mu
- building the sigma matrix by making ones(m,1) times sigma
- Defining M as X minus mu matrix
- Defining N as M element-wise divided by sigma matrix
- Renaming N into X (simple equal equation); it does not change anything as well if I rename N into X_norm, neither if I directly rename M element-wise divided by sigma matrix to be equal to X or to X_norm.

This is a complete puzzle to me.

YW Yiming Wang · 2 years ago

Hi Tom,

I have got to the step to derive the X_normalized vector. Please see the partial screen below. When I try to submit, I did not get "nice work" feedback. I wonder where could I have made a mistake in the featureNormalize. m?



```
Program paused. Press enter to continue.
Normalizing Features ...
Running gradient descent ...
Theta computed from gradient descent:
 0.000000
 0.000000
 0.000000

Predicted price of a 1650 sq-ft, 3 br house (using gradient descent):
 $0.000000
Program paused. Press enter to continue.
```

Also, when I want to look at the dimension of X_normalized, the Octave workspace doesn't respond to my request and returns nothing.

Thank you very much for your help in advance ! This discussion page has been super helpful thus far. I have tried to see if I have other problems raised by other folks in this discussions, but it seems not where my problem is.

Yiming

0 Upvotes Hide 9 Replies

TM Tom Mosher Mentor · 2 years ago

X_norm is a local variable inside the function. You can only inspect it if you put a breakpoint inside the function and then run the script or test case that uses the function.

The ex1_multi.m script saves the normalized X values back in the same variable name X

0 Upvotes

YW Yiming Wang · 2 years ago

Hi Tom,

Thank you for explaining that X_normalized values is saved as X. I still could not get my featureNormalized() to work, as my submission still did not pass. When I did the testcase for featureNormalized, my result is listed below, which is not correct. Could you please help me to understand why my featureNormailized function is not working? Thank you very much!

Yiming

```
>>[Xn mu sigma] = featureNormalize([1 2 3])

Xn =

1 2 3

mu = 2

sigma = 1
```

0 Upvotes

Tom Mosher Mentor · 2 years ago

That test case is not a very good one, and I would not worry about getting the correct results for that.

1 Upvote

YW Yiming Wang · 2 years ago

Hi Tom,

I did an experiment on my octave, if I copy all my code from featureNormalize() on to the Octave workspace (octave 3.8.0), and I call X_noramlized; my X_normalized data shows properly normalized values, but if I call featureNormalize(X), the answer doesn't show normalized data, rather shows the un-normalized data. Does this mean there is a bug in my featureNormalize.m function?

Yiming

1 Upvote

Tom Mosher Mentor · 2 years ago

Probably yes.

0 Upvotes

YW Yiming Wang · 2 years ago

Ok, thank, I fixed the bug:-) The submission passed.

0 Upvotes

Nicolas Kokel · 2 years ago

Hello,

I did as suggested here below:

```
X = [1 2 3; 4 5 6]          % creates a test matrix
mu = mean(X)                % returns a row vector
sigma = std(X)              % returns a row vector
m = size(X, 1)              % returns the number of rows in X
mu_matrix = ones(m, 1) * mu
sigma_matrix = ones(m, 1) * sigma
```

'Now you can subtract the mu matrix from X, and divide element-wise by the sigma matrix, and arrive at X_normalized.'

but the issue I am facing is that:

- the first colum of the mu_matrix is made of ones
- the first column of the sigma_matrix is made of zeros.
- by normalizing X for mean values and the first column is made of zeros since 1-1 = 0
- by normalizing X for mean values and then for sigma and the first colum is made of infinite values since it the result of 0 / 0.

Should the first colum of normalized X be made of zeros or of ones?

Does it means no normalization, or normalization only for mean values?

The way for doing so is by selectively normalizing columns 2 and 3 only?

1 Upvote

Tom Mosher Mentor · 2 years ago

Normalize the features before you add the bias units. ex1 does this for us.

0 Upvotes

Nicolas Kokel · 2 years ago

Oh! I found my mistake.

I was typing code in Octave AFTER runing ex1_multi, so that ones had already been added as the first of three columns to X.

When back to modify featureNormalized.m with MathLab first.

0 Upvotes

DK dinesh kuruba · 2 years ago · Edited

Can someone post test cases for NormalEqn() function ? and should we nomalize the parameters before calculating price ? Price calculated using gradient descent method and normalize equation does not match.

0 Upvotes Hide 1 Reply

DK dinesh kuruba · 2 years ago · Edited

Hi,

Sorry for the confusion as I used feature normalized vector for calculating the price using normalEquation method.

Thanks,

Dinesh

0 Upvotes

xiang zhou · 2 years ago

Hi Tom,

thank you for creating this section!

I understand the logic of feature normalisation , and say that now I have followed the steps and created the Matrix Xn which is the normalised Matrix of Matrix X.

What should I do next?

I am not sure what does the equation below in Ex1 Multi mean?

[X mu sigma] = featureNormalize(X);

Many thanks!

Erik

0 Upvotes Hide 4 Replies

Tom Mosher Mentor · 2 years ago · Edited

That line of code performs normalization on the X matrix, then returns it and saves it in the original "X" variable. It also provides you the mean (mu) and standard deviation (sigma) values for that data set.

You must use the mu and sigma values when you want to use the theta value to make a new prediction. See the instructions on Page 14 of ex1.pdf.

0 Upvotes

xiang zhou · 2 years ago

Thanks for your help Tom, it worked!:)

1 Upvote

ES Erick Emmanuel Pérez Solís · 2 years ago

Hello Tom, I am having a little problem, after perfoming feature normalization and running ex1_multi I check the values of X in the variable space and it doesn´t change its values, also I would like to know what is the function of the variable X_norm.

Thank you very much

0 Upvotes

Tom Mosher Mentor · 2 years ago

X_norm is a local variable inside the featureNormaliz() function. Its value is returned to the script that called the function.

ex1_multi.m saves this value back into the original "X" variable name.

0 Upvotes