

Week 3

Week 3

P

I dont even understand what you want us to do here!

parth [Assignment: Tobit Regression](#) · 3 days ago · Edited

Hi,

and I thought the first 2 assignments were cryptic...

don't you guys think this is a bit too much? you don't even show us what the formula you want us to implement looks like or what its supposed to do? you just want us to debug your notebook and figure it out? we are not mind readers we are here to learn.

I have been trying for hours but the matrix size of the inputs with bias and the weights just dont match, so what am i doing wrong?

self.out = tf.matmul(self.data_plus_bias, tf.transpose(self.weights))

dist = tf.contrib.distributions.Normal(1.0, 0.05)

term1= self.target * tf.log((1/self.sigma)* dist.prob(self.target-self.out/self.sigma))

term2=(1-self.target)*tf.log(1-dist.cdf(self.out-self.target/self.sigma))

are we supposed to add another column of ones or zeros to our input with bias ?

2 Upvotes

Reply

Follow this discussion

EarliestTopMost Recent

Matthieu Brucher · 5 hours ago

What wiki page? Seriously, there is a value L and no explanation as to what is what.

https://en.wikipedia.org/wiki/Tobit_model

0 Upvotes

Hide 1 Reply

Florian David Preis · 4 hours ago

That is the correct Wiki page. y_L is a censoring value, below which the data is cut off, i.e. set to 0. In our case y_L=0, so ignore it in the formula. y_i is the target, X_i beta is the output, (ly_i) is either 1 or 0 depending on y_i>0 or <0 respectively. \varphi is the pdf in our case distribution.Normal with the prob attribute, and \Phi is also a normal distribution with the attribute cdf (cumulative distribution function). Hope this helps.

0 Upvotes

Reply

Florian David Preis · 3 days ago

Hi,

- For your first formula: use the same as the one for self.output, i.e. with the slicing.
- In the argument of dist I used the same as in the second week's assignment in the MLE section.
- In term1 and term2 your use of self.target is wrong, check Wiki for tobit regression. It should be \theta(self.target) where \theta denotes the Heaviside theta function. The trick is to implement it in tensorflow. And don't forget the regulators! log(0) needs to be avoided.

1 Upvote

Hide 18 Replies

See earlier replies

Florian David Preis · 3 days ago

Hi,

check out the loss log_LL the MLE code in week 2 which was coded for us by the instructors. You need a regularization (add a very small but sufficiently large number) that prevents evaluating log(0). This should prevent the occurrence of nans.

Furthermore, in your term2 the argument of dist.cdf is incorrect. self.target does not appear there, only self.out does. Look carefully at the formula on wiki: y_L is not the target but the censoring value, which in our case is simply y_L=0.

You have yet more errors in term1 and term2: In the prefactor of both terms the indicator function ((ly) in wiki) appears there, while in the argument of dist.prob not the indicator function but the self.out itself should appear only.

0 Upvotes

RM

Ryland Mathews · 3 days ago

Hello,

Thank you for the guidance Florian! I am just a little confused. Is there material (or direction to material) where we are able to learn this? I have rewatched week 2 and week3 because I thought I may have missed something however, I am getting the feeling the class does not provide any guidance, and I am just meant to google the internet and read wikis to learn?

0 Upvotes

LV

Leo Volkov · 3 days ago · Edited

Check out term1 and term2 placeholder dims

in your realization term 1 and term2 have same dims as self.target, but they initialized completely different

... this is strange, but seems it doesn't influence on the result

0 Upvotes

P

parth · 3 days ago

hi, I know youre just trying to help but this isn't helping me get closer to my answer and im sure there are many people like me here, this is the first course ive experienced where they made it this hard to try and understand what they are trying to teach.

on wiki, the first term shows dist.prob (target - out / sigma)

0 Upvotes

P

parth · 3 days ago · Edited

these are my terms right now. am I any closer?

self.out = tf.matmul(self.data_plus_bias[:, :], self.weights[:-1])

dist = tf.distributions.Normal(loc=0.0, scale=1.0)

term1 = self.target * tf.log((1/self.sigma)* dist.prob((self.target - self.out)/self.sigma))

term2 = (1-self.target)*tf.log(0.00001*(1-dist.cdf((tf.sign(tf.nn.relu(self.out)))/self.sigma)))

0 Upvotes

RM

Ryland Mathews · 3 days ago

parth...these are mine. ift goes through no nans but its not correct

self.out = tf.matmul(self.data_plus_bias[:, :], self.weights[:-1])

dist = tf.distributions.Normal(loc=0.0, scale=1.0)

term1= tf.sign(self.target) * tf.log(((1/self.sigma)* dist.prob((self.target-self.out)/self.sigma)) +.00001)

term2=(1-tf.sign(self.target))*tf.log((1-dist.cdf((self.out)/self.sigma) + .00001)

0 Upvotes

P

parth · 3 days ago · Edited

yeah mine runs through as well but I cant get the right answer whatever that is I don't know even know where to start looking or how, this is so frustrating ive wasted hours on this

0 Upvotes

RM

Ryland Mathews · 3 days ago

self.out = tf.matmul(self.data_plus_bias[:, :], self.weights[:-1])

dist = tf.distributions.Normal(loc=0.0, scale=1.0)

term1= tf.sign(self.out) * tf.log(((1/self.sigma)* dist.prob((self.target-self.out)/self.sigma)) +.00001)

term2= (1-tf.sign(self.out)) * tf.log((1-dist.cdf((self.out)/self.sigma) + .00001)

this works and get no NANs . I been staring at the wiki and other forum posts. cant figure it out. Im trying to focus on what Leo said about our dims but im not sure what he means

0 Upvotes

Florian David Preis · 3 days ago

You are almost there. use inside the sign function a relu that acts on self.target not self.out and it should work.

0 Upvotes

P

parth · 3 days ago

I solved it now im stuck on the last question, again they don't even say what loss function to use

1 Upvote

RM

Ryland Mathews · 2 days ago

parth, florian. can you give me some guidance on the problem area. I am about to lose my mind haha. I made the changes Florian advised and still.. 0/100 on the assignment haha.

START CODE HERE ### (= 6-7 lines of code)

self.out = tf.matmul(self.data_plus_bias[:, :], self.weights[:-1])

dist = tf.distributions.Normal(loc=0.0, scale=1.0)

L_y = tf.sign(tf.nn.relu(self.target))

term1= L_y * tf.log(((1/self.sigma)* dist.prob(self.target-self.out/self.sigma)) +.00001)

term2=(1-L_y) * tf.log((1-dist.cdf((self.out)/self.sigma) + .00001)

END CODE HERE

self.loss = - tf.reduce_mean(term1 + term2)

0 Upvotes

RM

Ryland Mathews · 2 days ago

I think my L_y might not be coded properly?

0 Upvotes

P

parth · 2 days ago

nah, its all correct, the only thing you got wrong is the bracketing for the dist.cdf portion

0 Upvotes

RM

Ryland Mathews · 2 days ago

got it! there were mistakes with the parentheses.

0 Upvotes

P

parth · 2 days ago

now let me know if you can figure out the last part of the notebook.. haha I don't even know what they want there

0 Upvotes

Reply

Reply