Punto 1 Problem Set 3 ML

Octubre 15, 2021

Texto

Descripción generada automáticamente**BIG DATA AND MACHINE LEARNING FOR APPLIED ECONOMICS-2021-20**

**Profesor:** Ignacio Martin Sarmiento Barbieri

**Profesor :** Rafael Cano Polania

**Estudiantes:**

Hernando Mauricio Rodriguez cód. 201728877

1. **Binary Response Online Updating. The problem is simple but yet complicated. On-  
   line updating is essential because it breaks the storage barrier and helps with com-  
   putation. Assume that a new observation arrives, and instead of refitting the entire  
   model, you want to update your binary model estimates. Show that the contribution  
   made by the observation i to the likelihood function is**

**is globally concave with respect to β if the function F is such that F(−x) = 1 −F(x),  
and if its first derivative f, and its second derivative f′satisfy the condition:**

**for all real finite x. Show that this condition is satisfied by the logistic function**

**2. In many sub-fields of economics, like finance, it is common that the tails of the noise**

**distribution are much heavier than the standard Gaussian tails. One way to model**

**this is to use a t −distribution. Consider the following model: yi = α + βxi + ,i =**

**1,...,N,, i.e., a model with a constant and a single regressor. But now /σ ∼iid tv,**

**and E(X) = 0 and v are the degrees of freedom.**

**(a) Write down the log-likelihood, using the explicit formula for the density of the**

**t −distribution**

**(b) Find the first derivatives of this log-likelihood with respect to the parameters**

**α, β, σ2, and v.**

**(c) Assume that σ2 and v are known. Can you solve for the maximum likelihood**

**estimator of α and β? Do they match the least-square estimators? If not, why**

**and how do they differ?**

MLE

MLE

Least Square Estimators

Comparando los estimadores de MLE y MCO podemos ver que el estimador de es el mismo para las dos metodologías. Es decir que la forma de calcular la constante de nuestro modelo es el mismo. El estimador de si es diferente. En MLE se usa el promedio de las variables dependientes e independientes mientras que en MCO se usa el numero de observaciones para balancear el estimador.

**d) Using the software of your choice, write a function that takes as arguments some**

**data (y,x) and outputs the MLE estimates from a t −distribution.**

Se decidió usar Python y Jupyter el código que describe estas funciones es el siguiente. (Si se quiere ver más a fondo el código este se incluyó en el repositorio.)

#Ejercicio 1, Pregunta 2D

import operator

operatorlookup = {

'+': operator.add,

'-': operator.sub,

'\*': operator.mul,

'/': operator.truediv

}

#Se crean 2 listas de numeros para probar el codigo, esta lista se puede cambiar para incluir cualquier conjunto de numeros que se desee.

Setx=[2,3]

Sety=[10,50]

Meanx = sum(Setx)/len(Setx)

Meany = sum(Sety)/len(Sety)

#Se describe los estimadores que se encontraron en el enunciado pasado

Beta\_est=((np.dot(Setx,Sety))-Meany\*sum(Setx))/(np.dot(Setx,Setx)-Meanx\*sum(Setx))

Alpha\_est = Meany-Beta\_est\*Meanx

#Se regresan los valores de los estimadores

print("Alpha:" +str(Alpha\_est) +" Beta:" +str(Beta\_est))

**(e) Simulate some data that follow the model described above, and answer the**

**following questions:**

Debido a que el enunciado o el modelo original no menciona como se distribuye las variables independientes entonces se asume que esta distribuye de forma normal estándar:

#Ejercicio 1, Pregunta 2e

#Primero se tiene que simular el modelo

#Se simula la variable x con distribución normal

x = np.random.normal(0, 1, 100000)

#Como se dice en el modelo original, el error se distribuye de forma student t

e = np.random.standard\_t(10,size=100000)

#Se crea entonces la función y los estimadores alfa y beta se crean aleatoriamente

y = 2 + 5\*x + e

Simulacion = pd.DataFrame({'x':x, 'y':y})

df.head()

1. **How well does the MLE recover the parameters?**

#Ejercicio 1, Pregunta 2e\_i

#Se crean los datos al azar

x\_i = np.random.normal(0, 1, 100000 )

e\_i = np.random.standard\_t(10,size=100000 )

#Se plantea el modelo con los mismos datos de antes

y\_i = 2 + 5\*x\_i + e\_i

#Se hace el mismo proceso que en la pregunta anterior pero ahora la lista de numeros es simulada

Setx=x\_i.tolist()

Sety=y\_i.tolist()

Meanx = sum(Setx)/len(Setx)

Meany = sum(Sety)/len(Sety)

Beta\_est=((np.dot(Setx,Sety))-Meany\*sum(Setx))/(np.dot(Setx,Setx)-Meanx\*sum(Setx))

Alpha\_est = Meany-Beta\_est\*Meanx

#Se devuelven los estimadores por medio de MLE y se comparan con los que tenemos en el modelo simulado

print("Alpha:" +str(Alpha\_est) +" Beta:" +str(Beta\_est))

Alpha:2.0012182286588907 Beta:4.995456812434374

Los stimadores de MLE son muy parecidos a los que tenemos en el modelo simulado. Solo tienen una diferencia en centésima decimal. Podemos asumir que MLE es una muy buena metodología para conseguir estos estimadores.

1. **Does it get better as N grows?**

##Ejercicio 1, Pregunta 2e\_ii

#Mismo procedimiento que hemos llevado a cabo antes excepto que ahora el tamaño de la lista de numeros es 100 en vez de 100000

x\_ii = np.random.normal(0, 1, 100)

e\_ii = np.random.standard\_t(10,size=100)

y\_ii = 2 + 5\*x\_ii + e\_ii

Setx=x\_ii.tolist()

Sety=y\_ii.tolist()

Meanx = sum(Setx)/len(Setx)

Meany = sum(Sety)/len(Sety)

Beta\_est=((np.dot(Setx,Sety))-Meany\*sum(Setx))/(np.dot(Setx,Setx)-Meanx\*sum(Setx))

Alpha\_est = Meany-Beta\_est\*Meanx

print("Alpha:" +str(Alpha\_est) +" Beta:" +str(Beta\_est))

Alpha:2.2479875868155705 Beta:5.081704543860978

Comparando este resultado con el del enunciado anterior podemos ver que con un numero de observaciones más pequeños, los estimadores se separan más del modelo simulado. Con 100000 observaciones la diferencia absoluta entre los estimadores y la simulación es 0.0045 para Beta y 0.0012 para Alfa. Mientras tanto con solo 100 observaciones la diferencia es 0.081 para Beta y 0.247 para Alfa. Una diferencia bastante notable.

1. **What about when the variance of X increases?**

##Ejercicio 1, Pregunta 2e\_iii

# Se lleva a cabo el mismo procedimiento de antes pero ahora la varianza de nuestra variable pasa de ser 1 a 10000

x\_iii = np.random.normal(0, 10000, 10000)

e\_iii = np.random.standard\_t(10,size=10000)

y\_iii = 2 + 5\*x\_iii + e\_iii

Setx=x\_iii.tolist()

Sety=y\_iii.tolist()

Meanx = sum(Setx)/len(Setx)

Meany = sum(Sety)/len(Sety)

Beta\_est=((np.dot(Setx,Sety))-Meany\*sum(Setx))/(np.dot(Setx,Setx)-Meanx\*sum(Setx))

Alpha\_est = Meany-Beta\_est\*Meanx

print("Alpha:" +str(Alpha\_est) +" Beta:" +str(Beta\_est))

Alpha:2.005579368426112 Beta:4.999999161828031

Se cambia la varianza de la variable X de 1 a 10000. Este cambio grande se da para ver claramente si hay algún cambio en los estimadores. Dicho esto la diferencia entre el estimador de Beta y el modelo es 0.0000008, la diferencia de Alpha es 0.005. Podemos asumir que el cambio de la varianza de X no hace mucha diferencia en el momento de obtener los estimadores de forma MLE.

1. **How does it compare relative to a naive OLS estimator?**

#Ejercicio 1, Pregunta 2e\_iv

#La simulación de datos es igual a las situaciones pasadas

x\_iv = np.random.normal(0, 1, 10000)

e\_iv = np.random.standard\_t(10,size=10000)

y\_iv = 2 + 5\*x\_iv + e\_iv

Setx=x\_iv.tolist()

Sety=y\_iv.tolist()

Meanx = sum(Setx)/len(Setx)

Meany = sum(Sety)/len(Sety)

#Ahora se cambian las funciones de estimadores a las de MCO

Beta\_est=((len(x\_iv)\*(np.dot(Setx,Sety)))-(sum(Setx)\*sum(Sety)))/(len(x\_iv)\*((np.dot(Setx,Setx))-(sum(Setx)\*sum(Setx))))

Alpha\_est = Meany-Beta\_est\*Meanx

print("Alpha:" +str(Alpha\_est) +" Beta:" +str(Beta\_est))

Alpha:1.9863405550494948 Beta:6.086213090210755

Los estimadores de MCO tienen las diferencias más grandes entre los estimadores y la simulación que hemos visto hasta ahora. La diferencia absoluta de Beta es 1.086 y la diferencia de Alfa es 0.013. Debido a esto podemos concluir que los estimadores de MLE son mejores que los de MCO.

**(f) Each student in the class has an account in AWS. You can access it with your**

**@uniandes.edu.co user. Set up an AWS instance and install a software of your**

**choice. Attach screenshots of the virtual machine running. Some suggestions to**

**install**

**(g) Repeat the simulation in e using a parallel or distributed approach in AWS**

**i. Did you get the same results as above? If not, why?**

**ii. How did you handle the seed in this context?**

**iii. Was there a computational time gain? Report the differences.**

**iv.Indicate who in your team ran the simulations. (I’ll check AWS usage)**

Al tratar de ingresar a AWS con la cuenta de @uniandes.edu.co, esta no tiene registro de este usuario. Al tratar de crear un usuario con el correo de la institución la siguiente pagina aparece:

Graphical user interface

Description automatically generated

Nos encontramos sin acceso a una tarjeta de crédito para poder seguir con la creación de la cuenta. Por esta razón no se puede llevar a cabo las preguntas F o G.

**2.1 Getting to know Evanston, IL**

**This part of the problem set involves a series of spatial data sets on the City of Evanston,**

**IL. All the relevant data sets are in the data folder. The evanston\_parcel\_data.csv**

**file contains parcel level data from Evanston. The data was retrieved from the county**

**assessors’ office.1. The first objective is to showcase your mapping abilities. The second**

**objective is to use the tools studied in class to model and predict assessment values using**

**all the provided information**

**1. “Mapping the field”**

**(a) Begin by creating a map that includes census area identifiers (census blocks,**

**census tracts), major infrastructure layers (train line, roads, etc.), and Lake**

**Michigan shoreline**

Diagram

Description automatically generated with medium confidence

En el mapa anterior La línea verde representa la costa de la ciudad. Las líneas grises se usan como base para diferenciar los bloques de censo. Las líneas azules representan las carreteras principales. Las líneas rosadas y puntos morados representan las líneas de metro y las estaciones de este. Las líneas amarillas y los puntos amarillos representan las líneas de tren y las estaciones de este

**(b) Match the parcel data to the block level file and calculate (i) average assessment**

**values and (ii) building area to floor area at the block level**

Diagram

Description automatically generated

El mapa anterior mide el valor promedio de una propiedad por los bloques de censo. Entre mas se acerca al color amarillo más valioso son las propiedades en esa zona. Mientras tanto entre más purpura son los bloques menos valor tienen las propiedades.

**(c) Describe your results. In your description, include a side by side map using the**

**map you created in part (a) that includes the information you generated in (b)**

Comparando los dos mapas podemos notar claramente lo que hace un bloque de censo más valioso en promedio. La cercanía a el lago Michigan. Dicho esto la cercanía a los medios de transporte publico también ayuda, así sea muy poco, a que el valor de las propiedades aumente en promedio.