به نام خدا

تمرین درس یادگیری آماری

سری اول

امیدرضا داودنیا

پاییز۱۴۰۲

۱- بردار خطا e در رگرسیون خطی، به صورت زیر تعریف می‌شود:

$$\vec{e} = \vec{y} - \boldsymbol{X}\vec{\beta}$$

الف: نشان دهید $\boldsymbol{X}^T\vec{e} = 0$.

$$\vec{e} = \vec{y} - \boldsymbol{X}\vec{\hat{\beta}} \xrightarrow{\boldsymbol{X}^T\times(\dots)} \boldsymbol{X}^T \times (\vec{e}) = \boldsymbol{X}^T \times \left(\vec{y} - \boldsymbol{X}\vec{\hat{\beta}}\right) \rightarrow$$

$$\boldsymbol{X}^T\vec{e} = \boldsymbol{X}^T\vec{y} - \boldsymbol{X}^T\,\boldsymbol{X}\vec{\hat{\beta}} \xrightarrow{recall\ \boldsymbol{X}^T\vec{y} = \boldsymbol{X}^T\boldsymbol{X}\,\vec{\hat{\beta}}} \boldsymbol{X}^T\vec{e} = \boldsymbol{X}^T\boldsymbol{X}\,\vec{\hat{\beta}} - \boldsymbol{X}^T\,\boldsymbol{X}\vec{\hat{\beta}} = 0$$

درصورتی که بتای اولی (سمت چپ) بتای واقعی نیز باشد رابطه بالا برقرار است چرا که با امید گیری نتیجه مشابه حاصل خواهد شد.

ب: نشان دهید $\vec{\hat{y}}^T\vec{e} = 0$.

با استفاده از نتیجه قسمت قبلی داریم ؛ $\boldsymbol{X}^T\vec{e} = 0$ به همین صورت داریم ؛

$$\vec{\hat{y}}^T\vec{e} = \vec{e}\vec{\hat{y}} = \vec{e}\left(\boldsymbol{X}\vec{\hat{\beta}}\right) = (\boldsymbol{X}^T\vec{e})\,\vec{\hat{\beta}} \xrightarrow{inside\ Parentheses\ is\ 0} \vec{\hat{y}}^T\vec{e} = 0$$

۲- در صورتیکه داده های آموزش را در ماتریس مربعی و معکوس پذیر $\boldsymbol{Q}$ ضرب کنیم $(\boldsymbol{D} = \boldsymbol{XQ})$ ، نشان دهید بردار خطا در رگرسیون خطی در دوحالتی که از $\boldsymbol{D}$ و $\boldsymbol{X}$ برای تخمین $\vec{y}$ استفاده کنیم یکسان است. یعنی:

$$\vec{e} = \vec{y} - \boldsymbol{X}\vec{\hat{\beta}} = \vec{y} - \boldsymbol{D}\overrightarrow{\hat{\beta}'}$$

از رابطه سمت چپ داریم ؛

$$\boldsymbol{X}^T\vec{e} = \boldsymbol{X}^T\left[\vec{y} - \boldsymbol{D}\vec{\hat{\hat{\beta}}}\right] = 0 \xrightarrow[\boldsymbol{X}^T\vec{e}=0]{} \vec{\hat{\beta}} = \boldsymbol{Q}\vec{\hat{\hat{\beta}}}$$

$$\vec{e} = \vec{y} - \boldsymbol{X}\vec{\hat{\beta}} \xrightarrow{\boldsymbol{DQ}^{-1} = \boldsymbol{X}\ and\ \vec{\hat{\beta}} = \boldsymbol{Q}\vec{\hat{\hat{\beta}}}} \vec{e} = \vec{y} - (\boldsymbol{DQ}^{-1})\left(\boldsymbol{Q}\vec{\hat{\hat{\beta}}}\right) \xrightarrow{\boldsymbol{Q}^{-1}\boldsymbol{Q} = I} \vec{e} = \vec{y} - \boldsymbol{D}I\vec{\hat{\hat{\beta}}}$$

۳- فرض کنید مقدار بهینه $\vec{\beta}$ با استفاده از یک مجموعه داده محاسبه شده است. حال میخواهیم یک نمونه جدید $(\vec{x}_a, y_a)$ را به داده های آموزش اضافه کنیم. نشان دهید مقدار بهینه جدید $\vec{\beta}_{new}$ از رابطه زیر بدست می‌آید.

$$\vec{\beta}_{new} = \vec{\beta} + \frac{1}{1 + \vec{x}_a{}^T(\boldsymbol{X}^T\boldsymbol{X})^{-1}\vec{x}_a}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\vec{x}_a(y_a - \vec{x}_a{}^T\vec{\beta})$$

از آنجایی که در راهنمایی رنک ماتریس B برابر با ۱ و این ماتریس مربعی معکوس پذیر است ، پس این ماتریس نمی تواند تکینگی داشته باشد و این ماتریس یک عدد است ، به همین ترتیب چون این عدد B با

ماتریس A جمع شده است پس ماتریس A نیز عدد است ، پس احتمالا مقصود مساله حل مساله برای رگرسیون تک متغییره باید باشد.

اگر ماتریس داده قدیمی را بدین صورت تعریف کنیم.

$$\boldsymbol{X}^T = [\begin{matrix} 1 & \dots & x_1^{(1)} \\ 1 & \dots & x_1^{(n)} \end{matrix}]$$

به همین صورت ماتریس مربعی ضرب ترانهاده این ماتریس در خودش بدین صورت معرفی خواهد شد ؛

$$\boldsymbol{X}^T\boldsymbol{X} = [\begin{matrix} \boldsymbol{n} & \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} x_1^{(i)} \\ \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} x_1^{(i)} & \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} (x_1^{(i)} x_1^{(i)}) \end{matrix}]$$

$$\beta_n - \beta = \frac{(\boldsymbol{X}^T\boldsymbol{X})^{-1} x_a y_a - (\boldsymbol{X}^T\boldsymbol{X})^{-1} x_a x_a^T \beta}{1 + x_a^T (\boldsymbol{X}^T\boldsymbol{X})^{-1} x_a}$$

باتوجه به داده های صورت سوال ، همچنین یک فرض کوچک که فعلا بتا صفر نداریم؛ به ادامه حل مسئله می پردازیم ، در این صورت سمت راست مسئله باید عدد اسکالر شود.

به همین صورت مخرج کسر نیز که با عدد یک اسکار جمع شده است نیز سمت راستش باید اسکالر باشد؛ با توجه به ابعاد مسئله خواهیم داشت.

$$\beta_n = (\boldsymbol{X}^T\boldsymbol{X})^{-1} \boldsymbol{X}^T \boldsymbol{Y}$$

اگر داشته باشیم

$$\frac{d\beta_n}{dx_a} = \frac{\beta_n - \beta}{\Delta x} = \frac{(\boldsymbol{X}^T\boldsymbol{X})^{-1} x_a (y_a - \hat{y}_a)}{1 + x_a^T (\boldsymbol{X}^T\boldsymbol{X})^{-1} x_a} \rightarrow \beta_n - \beta = \frac{(\boldsymbol{X}^T\boldsymbol{X})^{-1} x_a y_a - (\boldsymbol{X}^T\boldsymbol{X})^{-1} x_a x_a^T \beta}{1 + x_a^T (\boldsymbol{X}^T\boldsymbol{X})^{-1} x_a}$$

$$\beta_n = \beta + \frac{(\boldsymbol{X}^T\boldsymbol{X})^{-1} x_a y_a - (\boldsymbol{X}^T\boldsymbol{X})^{-1} x_a x_a^T \beta}{1 + x_a^T (\boldsymbol{X}^T\boldsymbol{X})^{-1} x_a}$$

۴- با در نظر گرفتن ورودی و خروجی به صورت زیر و به کمک رگرسیون خطی یک متغیره، $\vec{\hat{\beta}}$ و تخمین واریانس نویز ($\hat{\sigma}^2$) را محاسبه کنید.

X	Y
2	19.73
3	26.94
4	35.71

فرمول رگرسیون خطی یک متغییره را بدین صورت درنظر میگیریم ، $\hat{y} = \beta_0 + \beta_1 x$ در این صورت برای محاسبه پارامتر ها خواهیم داشت ؛با استفاده از کد زیر محاسبات انجام شده اند،در مثال بتاها منظور بتا هت هستند.

```
import numpy as np
n = 3
p = 1
x = np.array([2,3,4])
y =  [19.73,26.94,35.71]
y = np.array(y)
xBar = np.mean(x)                                          #3.0
yBar = np.mean(y)                                          #27.459999999999997
x = np.array([2,3,4])
xZeroMean = x - xBar                                       #array([-1.,  0.,  1.])
yZeroMean = y - yBar
#array([-7.73, -0.52,  8.25])
beta1 = np.sum(xZeroMean*yZeroMean) / np.sum(xZeroMean*xZeroMean) #7.99
beta0 = yBar - beta1*xBar                                  #3.4899999999999984
yHat = beta0 + beta1*x
#array([19.47, 27.46, 35.45])
rss = np.sum((y-yHat)*(y-yHat))                            #0.40559999999999563
varHat = rss /(n-p-1)                                      #0.40559999999999563
```

$$\bar{x} = 3 \; and \; \bar{y} = 27.46$$

$$\beta_1 = \frac{< x - \bar{x}. y - \bar{y} >}{< x - \bar{x}. x - \bar{x} >} = 7.99$$

$$\beta_0 = \bar{y} - \beta_1 \bar{x} = 3.49$$

$$\hat{y} = \beta_0 + x\beta_1 = [19.47 \quad 27.46 \quad 35.45]^T$$

$$RSS = (y - \hat{y})^T (y - \hat{y}) = 0.406$$

$$\hat{\sigma}^2 = \frac{1}{n-p-1} RSS = 0.406$$

۵- فرض کنید داده آموزشی زیر را در اختیار داریم:

X	Y
1	3
2	1
3	0.5

و بخواهیم از مدل $\hat{y} = \hat{a}x^{\hat{b}}$ برای تخمین رابطه ی ورودی و خروجی استفاده کنیم. با استفاده رگرسیون خطی، ضرایب $\hat{a}$ و $\hat{b}$ را محاسبه کنید.
(**راهنمایی**: از طرفین معادله $\hat{y} = \hat{a}x^{\hat{b}}$، لگاریتم بگیرید.)

با استفاده از تغییر متغییر داده شده خواهیم داشت ؛در حل سوال بتاها منظور بتا تخمین زده شده هستند.

$$\hat{y} = \hat{a}x^{\hat{b}} \xrightarrow{\log(.=.)} \log(\hat{y}) = \log(\hat{a}x^{\hat{b}}) \rightarrow \log(\hat{y}) = \log(\hat{a}) + \hat{b}\log(x)$$

حال این مسئله را به فرمی از رگرسیون خطی که می شناسیم برای سادگی بازنویسی خواهیم کرد.

$$y' = \log(\hat{y}) \;\; and \;\; \beta_0 = \log(\hat{a}) \;\; and \;\; \beta_1 = \hat{b} \text{ and } \acute{x} = \log(x)$$

برای حل سوال از تکه کد زیر استفاده شده است.

```python
# Import the numpy library and give it an alias 'np'
import numpy as np

# Create numpy arrays x and y
x = np.array([1,2,3])
y = np.array([3,1,0.5])

# Compute the natural logarithm of x and y and store the results in xPrime and yPrime
xPrime = np.log(x) # array([0.        , 0.69314718, 1.09861229])
yPrime = np.log(y) # array([ 1.09861229,  0.        , -0.69314718])

# Compute the mean (average) of xPrime and yPrime and store the results in xBar and yBar
xBar = np.mean(xPrime) # 0.5972531564093516
yBar = np.mean(yPrime) # 0.13515503603605483

# Subtract the mean from xPrime and yPrime and store the results in xZeroMean and yZeroMean
xZeroMean = xPrime - xBar # array([-0.59725316,  0.09589402,  0.50135913])
yZeroMean = yPrime - yBar # array([-0.59725316,  0.09589402,  0.50135913])

# Calculate the beta1 coefficient using linear regression formulas
beta1 = np.sum(xZeroMean * yZeroMean) / np.sum(xZeroMean * xZeroMean) # -1.6259799061924656

# Calculate the beta0 coefficient using linear regression formulas
beta0 = yBar - beta1 * xBar # 1.1062766672676863

# Calculate the predicted values of y (yHatPrime) based on the linear regression model
yHatPrime = beta0 + beta1 * xPrime # array([ 1.10627667, -0.02076672, -0.68004484])
RSSPrime       = np.sum((yPrime-yHatPrime)*(yPrime-yHatPrime)) #0.0006616707332145586

# Calculate the exponent of yHatPrime to get the final predicted values of y (yHat)
yHat = np.exp(yHatPrime) # array([3.02308148, 0.97944742, 0.50659428])
RSS  = np.sum((y-yHat)*(y-yHat)) #0.000998647411109344
a    = np.exp(beta0)            #3.02308147539285
b    = beta1                    #-1.6259799061924656
```

برای اعمال تغییر متغییر داده شده جدول داده های ما نیز بدین صورت تغییر خواهد کرد.

$\acute{x}$	$\acute{y}$
0	1.1
0.69	0
1.1	-0.69

مشابه سوالی قبلی خواهیم داشت.

$$\beta_1 = \frac{< x-\bar{x}.y-\bar{y} >}{< x-\bar{x}.x-\bar{x} >} = -1.62$$

$$\beta_0 = \bar{y} - \beta_1\bar{x} = 1.1$$

$$y' = [1.10627667 \quad -0.02076672 \quad -0.68004484]^T$$

$$\hat{y} = [3.02308148 \quad 0.97944742 \quad 0.50659428]^T$$

$R\acute{S}S = 0.00066167073321455586$

$RSS = 0.000998647411109344$

$\hat{a} \quad = 3.02308147539285$

$\hat{b} \quad = -1.6259799061924656$

سوال شبیه سازی در فایل نوت بوک به تفصیل توضیح داده شده و به خوبی پیاده سازی شده است.