19T2: COMP9417 Machine Learning and Data Mining

Lecture(s): Regression
Content: Review; Questions on topics in lecture
Version: With answers

Review

R1

Obtain the partial derivatives with respect to one value, when

$$f(x, y) = a_1 x^2 y^2 + a_4 xy + a_5 x + a_7$$

Answer

$$\frac{\partial f(x, y)}{\partial x} = 2x(a_1 y^2) + a_4 y + a_5$$

$$\frac{\partial f(x, y)}{\partial y} = 2y(a_1 x^2) + a_4 x$$

R2

When

$$f(x, y) = a_1 x^2 y^2 + a_2 x^2 y + a_3 xy^2 + a_4 xy + a_5 x + a_6 y + a_7$$

what will $\frac{\partial f(x,y)}{\partial x}$ and $\frac{\partial f(x,y)}{\partial y}$ be?

Answer

$$\frac{\partial f(x,y)}{\partial x} = 2a_1xy^2 + 2a_2xy + a_3y^2 + a_4x + a_5$$

$$\frac{\partial f(x,y)}{\partial x} = 2a_1x^2y + a_2x^2 + 2a_3xy2 + a_4x + a_6$$

R3

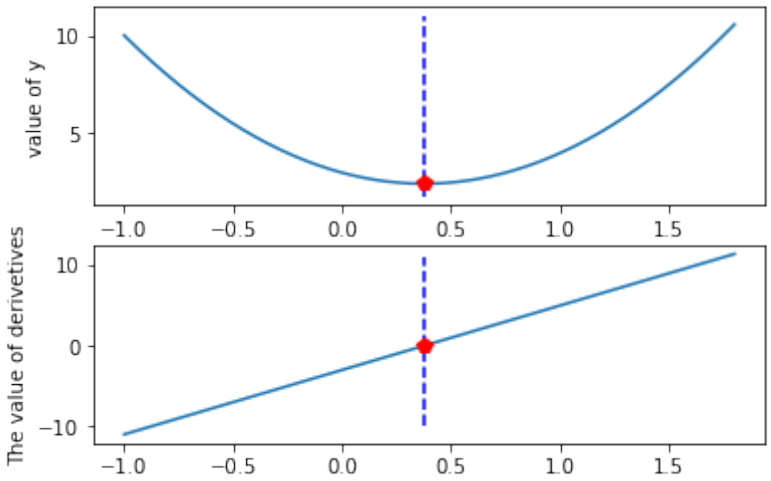
Do you recall how to solve optimization problems for Quadratic function? For example, $y = 4x^2 - 3x + 3$. Is the solution a minimum or maximum?

Answer

$$\frac{\partial y}{\partial x} = 8x - 3$$

when $x = \frac{3}{8}, \frac{\partial y}{\partial x} = 0$ and y is minimum.

Hint:



R4

What is the *loss function* for linear regression?

Answer

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{n}(y_i - (w_0 + w_1 x_i))^2$$

.

Q1

Go through this derivation and complete the exercise at the end of it.

A *univariate linear regression model* is a linear equation $y = a+bx$. Learning such a model requires fitting it to a sample of training data so as to minimize the error function $\mathcal{L} = \sum_{i=1}^{n}(y_i-(w_0+w_1x_i))^2$. To find the best parameters a and b that minimize this error function we need to find the error *gradients* $\frac{\partial \mathcal{L}}{\partial w_0}$ and $\frac{\partial \mathcal{L}}{\partial w_1}$. So we need to derive these expressions by taking partial derivatives, set them to zero, and solve for w_0 and w_1.

First we write the loss function for the univariate linear regression $y = w_0 + w_1 x$ as

$$\begin{aligned}
\mathcal{L} &= \frac{1}{N}\sum_{n=1}^{N}(y_n - (w_0 + w_1 x_n))^2 \\
&= \frac{1}{N}\sum_{n=1}^{N}(y_n - (w_0 + w_1 x_n))(y_n - (w_0 + w_1 x_n)) \\
&= \dots \\
&= \frac{1}{N}\sum_{n=1}^{N}[w_1^2 x_n^2 + 2w_1 x_n(w_0 - y_n) + w_0^2 - 2w_0 y_n + y_n^2]
\end{aligned}$$

At a minimum of $\mathcal{L}$ the partial derivatives with respect to w_0, w_1 should be zero. We will start with w_0, so first we remove from the above expression all terms not including w_0.

$$\frac{1}{N}\sum_{n=1}^{N}[w_0^2 + 2w_1 x_n w_0 - 2w_0 y_n]$$

Rearrange, taking terms not indexed by n outside:

$$w_0^2 + 2w_0 w_1 \frac{1}{N}(\sum_{n=1}^{N} x_n) - 2w_0 \frac{1}{N}(\sum_{n=1}^{N} y_n)$$

Taking the partial derivative with respect to w_0 we get:

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial w_0} &= 2w_0 + 2w_1 \frac{1}{N}(\sum_{n=1}^{N} x_n) - \frac{2}{N}(\sum_{n=1}^{N} y_n) \\
E(y) &= \frac{\sum_{n=1}^{N} y_n}{N} \\
E(x) &= \frac{\sum_{n=1}^{N} x_n}{N} \\
\widehat{w_0} &= E(y) - w_1 E(x)
\end{aligned}$$

Now we do the same for w_1, first removing all terms not including w_1:

$$\frac{1}{N}\sum_{n=1}^{N}[w_1^2 x_n^2 + 2w_1 x_n w_0 - 2w_1 x_n y_n]$$

Rearrange, taking terms not indexed by n outside:

$$w_1^2 \frac{1}{N}(\sum_{n=1}^{N} x_n^2) + 2w_1 \frac{1}{N}(\sum_{n=1}^{N} x_n(w_0 - y_n))$$

Taking the partial derivative with respect to w_1 we get:

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial w_1} &= 2w_1 \frac{1}{N}(\sum_{n=1}^{N} x_n^2) + \frac{2}{N}(\sum_{n=1}^{N} x_n(w_0 - y_n)) \\
&= w_1 \frac{2}{N}(\sum_{n=1}^{N} x_n^2) + \frac{2}{N}(\sum_{n=1}^{N} x_n(\widehat{w_0} - y_n)) \\
&= w_1 \frac{2}{N}(\sum_{n=1}^{N} x_n^2) + \frac{2}{N}(\sum_{n=1}^{N} x_n(E(y) - w_1 E(x) - y_n)) \\
&= \dots \\
&= 2w_1[\frac{1}{N}(\sum_{n=1}^{N} x_n^2) - E(x)E(x)] + 2E(y)E(x) - 2\frac{1}{N}(\sum_{n=1}^{N} x_n y_n) \\
&= 2w_1 E(x^2) - 2w_1 (E(x))^2 + 2E(y)E(x) - 2E(xy)
\end{aligned}$$

$$\frac{\partial \mathcal{L}}{\partial w_1} = 0$$

$$w_1(E(x^2) - (E(x))^2) = E(xy) - E(y)E(x)$$

$$
\begin{aligned}
\widehat{w_1} &= \frac{E(xy) - E(x)E(y)}{E(x^2) - (E(x))^2} \\
&= \frac{E(xy) - E(x)E(y)}{Var(x)} \\
&= \frac{Cov(x, y)}{Var(x)} \\
\widehat{w_0} &= E(y) - w_1 E(x)
\end{aligned}
$$

Exercise: To make sure you know the process, try to solve the following loss function for linear regression with a version of "$L2$" regularization:

$$\mathcal{L} = \frac{1}{N} \sum_n (y_n - (w_0 + w_1 x_n))^2 + \frac{\lambda (w_1)^2}{2}$$

Answer

$$
\begin{aligned}
\widehat{w_0} &= E(y) - w_1 E(x) \\
\widehat{w_1} &= \frac{E(xy) - E(x)E(y)}{E(x^2) - (E(x))^2 + \lambda}
\end{aligned}
$$

Q2

An intuitive understanding of the *regression coefficient* w_1 for univariate regression is that it defined as:

$$\frac{\text{covariance of } x \text{ and } y}{\text{variance of } x}$$

and a straightforward, if inefficient, way to compute this is:

$$w_1 = \frac{\Sigma(x - \bar{x})(y - \bar{y})}{\Sigma(x - \bar{x})^2}$$

where $\bar{v}$ represents the mean of the values in the dataset for variable v. Once w_1 is obtained we can find $w_0 = \bar{y} - w_1 \bar{x}$. Apply this method to determine the linear regression equation $y = w_0 + w_1 x$ for the small dataset below.

x	y
3	13
6	8
7	11
8	2
11	6

However, the same univariate regression can be written in matrix notation as

$$\widehat{\mathbf{w}} = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}$$

(this expression is used for multivariate linear regression, but it also applies to univariate linear regression using homogeneous coordinates).

We can see that this expression essentially is the variance of x represented by $(\mathbf{X}^{\mathrm{T}}\mathbf{X})$) for which we take the inverse, multiplied by the covariance of x and y – in other words, it is the same expression as we had before.

Now apply this expression to derive the vector of estimated coefficients, $\widehat{\mathbf{w}}$ to the dataset above. First, you will need to recall the definition of the inverse of a 2×2 matrix, available at many places on the web, e.g., `http://mathworld.wolfram.com/MatrixInverse.html`.

For a 2×2 matrix (why is $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ a 2×2 matrix?) it is possible (and possibly instructive) to calculate out the matrix operations by hand, including the inversion, but you will find it easier to put the x and y data into a matrix and vector representation and do the calculation in NumPy (or some alternative such as Matlab).

You should, of course, find the same values using both methods, and this example is sufficiently simple to intuitively see what the coefficients should be.

Answer

1. Calculate the mean of X and Y

$$\begin{aligned}
\bar{x} &= \sum_i (x_i) \\
&= \frac{3+6+7+8+11}{5} \\
&= 7 \\
\bar{y} &= \sum_i (y_i) \\
&= \frac{13+8+11+2+6}{5} \\
&= 8
\end{aligned}$$

2. Calculate the Cov(X,Y) and Var(X)

$$\begin{aligned}
Cov(X,Y) &= \frac{\sum_i((x_i-\bar{x})(y_i-\bar{y}))}{N} \\
&= \frac{\sum_i(x_i*y_i)}{N} - \bar{x}\bar{y} \\
&= \frac{3*13+6*8+7*11+8*2+11*6}{5} - 7*8 \\
&= -6.8 \\
Var(X) &= \frac{\sum_i((x_i-\bar{x})^2)}{N} \\
&= \frac{\sum_i(x_i^2)}{N} - \bar{x}^2 \\
&= \frac{3^2+6^2+7^2+8^2+11^2}{5} - 7*7 \\
&= 6.8
\end{aligned}$$

3. Calculate w_1, w_0 based on $\widehat{w_1} = \frac{COV(x,y)}{Var(x)}$ and $\widehat{w_0} = E(y) - w_1 E(x)$

$$\begin{aligned}
w_1 &= \frac{-6.8}{6.8} \\
&= -1 \\
w_0 &= \bar{y} - w_1 * \bar{x} \\
&= 8 - (-1) * y \\
&= 15
\end{aligned}$$

The result linear regression should be y=15-x

However, one way to solve this is to set up the matrix $\mathbf{X}$ using homogeneous coordinates, and vector $\mathbf{y}$, and simply compute the matrix product, invert it and complete the multiplication.

$$\mathbf{X} = \begin{pmatrix} 1 & 3 \\ 1 & 6 \\ 1 & 7 \\ 1 & 8 \\ 1 & 11 \end{pmatrix}$$

So

$$\begin{aligned}\mathbf{X}^{\mathrm{T}}\mathbf{X} &= \begin{pmatrix}1 & 1 & 1 & 1 & 1\\ 3 & 6 & 7 & 8 & 11\end{pmatrix}\begin{pmatrix}1 & 3\\ 1 & 6\\ 1 & 7\\ 1 & 8\\ 1 & 11\end{pmatrix}\\ &= \begin{pmatrix}5 & 35\\ 25 & 279\end{pmatrix}\end{aligned}$$

By calculating the inverse, we have

$$(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1} = \begin{pmatrix}\frac{279}{170} & -\frac{7}{34}\\ -\frac{7}{34} & \frac{1}{34}\end{pmatrix}$$

and

$$\begin{aligned}\mathbf{X}^{\mathrm{T}}\mathbf{y} &= \begin{pmatrix}1 & 1 & 1 & 1 & 1\\ 3 & 6 & 7 & 8 & 11\end{pmatrix}\begin{pmatrix}13\\ 8\\ 11\\ 2\\ 6\end{pmatrix}\\ &= \begin{pmatrix}40\\ 246\end{pmatrix}\end{aligned}$$

Therefore,

$$\widehat{\mathbf{w}} = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y} = \begin{pmatrix}15 & -1\end{pmatrix}$$

Here is some Python code.

```
import numpy as np
from numpy.linalg import inv

x = np.matrix([[1,3],[1,6],[1,7],[1,8],[1,11]])
y = np.array([13,8,11,2,6])

xtxi = inv(np.matmul(np.transpose(x),x))
xtxixt = np.matmul(xtxi,np.transpose(x))
coefficients = np.matmul(xtxixt,y)
print(coefficients)
```
