# 18s1: COMP9417 Machine Learning and Data Mining

**Lectures**: Linear Models for Regression
**Topic**: Questions from lecture topics
**Version**: with answers

**Last revision**: Mon Mar 19 13:05:55 AEDT 2018

## Introduction

Some questions and exercises from the course lectures covering aspects of learning linear models (models "linear in the parameters") for regression, i.e., numeric prediction, tasks.

**Question 1** *NOTE: Since a number of people have asked about the derivation of the coefficients in the lecture, here it is again. The point of including it in this tutorial is to be able to review the derivation if you wish, and to set up the following questions.*

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 $y = w_0 + w_1x$ as

$$\begin{aligned}
\mathcal{L} &= \frac{1}{N}\sum_{n=1}^{N}(y_n - (w_0 + w_1x_n))^2 \\
&= \frac{1}{N}\sum_{n=1}^{N}(y_n - (w_0 + w_1x_n))(y_n - (w_0 + w_1x_n)) \\
&= \dots \\
&= \frac{1}{N}\sum_{n=1}^{N}[w_1^2x_n^2 + 2w_1x_n(w_0 - y_n) + w_0^2 - 2w_0y_n + y_n^2]
\end{aligned}$$

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

$$\frac{1}{N}\sum_{n=1}^{N}[w_1^2x_n^2 + 2w_1x_nw_0 - 2w_1x_ny_n]$$

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:

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

Now we do the same for $w_0$, first removing 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:

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

We now have expressions for the partial derivative of the loss function with respect to $w_0$ and $w_1$. To find expressions for the minimum of the loss function we need to set each expression to zero and solve, starting with $w_0$:

$$\begin{aligned} 0 &= 2w_0 + 2w_1 \frac{1}{N}(\sum_{n=1}^{N} x_n) - \frac{2}{N}(\sum_{n=1}^{N} y_n) \\ 2w_0 &= \frac{2}{N}(\sum_{n=1}^{N} y_n) - w_1 \frac{2}{N}(\sum_{n=1}^{N} x_n) \\ w_0 &= \frac{1}{N}(\sum_{n=1}^{N} y_n) - w_1 \frac{1}{N}(\sum_{n=1}^{N} x_n) \end{aligned}$$

Representing the terms for the averages $\frac{1}{N}(\sum_{n=1}^{N} y_n)$ as $\bar{y}$ and $\frac{1}{N}(\sum_{n=1}^{N} x_n)$ as $\bar{x}$ we obtain:

$$\boxed{\widehat{w_0} = \bar{y} - w_1 \bar{x}}$$

One thing this expression tells us is that the average value of the output (the regression function) is defined in terms of the average value of the input variable $x$. This estimate for $w_0$ can now be substituted into the expression obtained above for the partial derivative with respect to $w_1$:

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

We can use some notation to simplify this expression (we skipped some steps in the derivation). Replacing $\frac{1}{N}(\sum_{n=1}^{N} x_n^2)$ with $\overline{x^2}$ and $\frac{1}{N}(\sum_{n=1}^{N} x_n y_n)$ with $\overline{xy}$ and setting to zero we can get:

$$\boxed{\widehat{w_1} = \frac{\overline{xy} - \bar{x}\bar{y}}{\overline{x^2} - (\bar{x})^2}}$$

Looking back at the lecture notes we can see that the denominator of this expression is the *variance* of the independent variable $x$ (Slide 31) and the numerator is the *covariance* of the dependent variable $y$ and $x$ (Slide 38). Compare the expressions for the intercept $w_0$ and the regression coefficient $w_1$ we have found with those that are derived for the quantities $a$ and $b$ on Slide 54 for the equation relating people's height and weight. You should be able to see that they are the same!

**Question 2** A linear regression model is represented by the linear equation $y = a + bx$. Show that the mean point $(\bar{x}, \bar{y})$ must line on the regression line.

---

**Answer**

In Question 1 we derived the expression $a = \bar{y} - b\bar{x}$. Suppose we ask the question: what is the value of the output $y$ for the linear regression model when the input is $\bar{x}$, i.e., the mean of $x$ ?

We formulate the regression model like this, then substitute the expression for $a$:

$$
\begin{aligned}
y &= a + b\bar{x} \\
&= \bar{y} - b\bar{x} + b\bar{x} \\
&= \bar{y}
\end{aligned}
$$

which completes the derivation.

---

**Question 3** We are told in the lecture notes that the sum of the residuals of the least-squares solution is zero. Complete the steps in the derivation shown on Slide 60 of the lecture notes:

$$\sum_{i=1}^{n}(y_i - (\hat{a} + \hat{b}x_i)) = 0$$

.

What is one consequence of this property that you need to be aware of when applying Linear Regression to real data?

---

**Answer**

Show that the sum of residuals of a linear regression function is zero.

$$\begin{aligned}
\sum_{i=1}^{n}(y_i - (\hat{a} + \hat{b}x_i)) &= \\
\sum_{i=1}^{n}(y_i - \hat{a} - \hat{b}x_i) &= \\
(\sum_{i=1}^{n} y_i) - n\hat{a} - n\hat{b}(\sum_{i=1}^{n} x_i) &= \\
n\bar{y} - n\hat{a} - n\hat{b}\bar{x} &= \\
n(\bar{y} - \hat{a} - \hat{b}\bar{x}) &= 0
\end{aligned}$$

A consequence is that regression will be susceptible to outliers, since it must fit the line to minimise the mean distance to all points and balance the residuals by summing to zero – outliers may "pull" the regression line, thereby distorting it wrt the rest of the data.

---

**Question 4** 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 on Slide 71 of the lecture notes, 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, $\hat{\mathbf{w}}$ to the dataset above. First, you will need to recall the definition of the inverse of a $2 \times 2$ matrix, available at many places on the web, e.g., `http://mathworld.wolfram.com/MatrixInverse.html`.

For a $2 \times 2$ matrix (why is $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ a $2 \times 2$ 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**

If you simply work through the method taking means of each variable, subtracting the values from their respective means, then completing the computation of covariance (for $x$ and $y$) and variance (for $x$) you should get:

$$w_1 = \frac{-34}{34} = -1$$

$$w_0 = \bar{y} - w_1\bar{x} = 15$$

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

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

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)
```

---