Linear regression notes

John Bogovic

2019 July

1 Review

To be multiplied, matrices need to have the same "inner" dimensions: the number of columns of the left matrix must equal the number of rows of the right matrix.

$$\boldsymbol{A}_{L\times M}\boldsymbol{B}_{M\times N} = \boldsymbol{X}_{L\times N} \tag{1}$$

but

$$\boldsymbol{A}_{L\times M}\boldsymbol{B}_{N\times M} = \text{nothing, nonsense} \tag{2}$$

This works:

$$\boldsymbol{A}_{L\times M}\boldsymbol{B}^T_{N\times M} = \boldsymbol{X}_{L\times N} \tag{3}$$

1.1 Exercise

If $\boldsymbol{A}_{M\times N}$, what size is $\boldsymbol{A}^T\boldsymbol{A}$?

1. $N \times N$
2. $M \times N$
3. $M \times M$
4. Stupid question, because you can't multiply $\boldsymbol{A}$ and $\boldsymbol{A}^T$.

2 Simple linear regression (fit a 1d line)

2.1 The problem

We are given the value of a variable, x, we would like to predict the value of another variable y. You will often see x called the "independent variable",

and y called the "dependent variable". We have many pairs of observations:

$$\begin{aligned} &x_1, y_1 \\ &x_2, y_2 \\ &\vdots \\ &x_N, y_N \end{aligned} \tag{4}$$

Linear regression (1d) does this by finding the linear function that gives the best predictions. The functions we have to consider are:

$$\hat{y} = ax + b \tag{5}$$

Where we wrote $\hat{y}$ instead of y to indicate that it is an estimate, or prediction, and not the true value of y for the given x. Another way to think of the task is that we need to find the values a and b that give us the best results. The values a and b are called "parameters" of the function. How do we measure how good the predictions are?

2.2 "Cost function" - measuring goodness of the prediction

The most common way is to use the "sum of squared differences" (SSD) also called "sum of squared errors" (SSE), or"residual sum of squares." It is computed like this:

$$SSD(a, b) = \sum_i (y_i - (ax_i + b))^2. \tag{6}$$

Notice that $ax_i + b$ is the value of y predicted by our function for the input x_i. This is an *"ordinary" linear least squares* problem. Figure 1 shows a visualization of SSD.

A related measure, the Root mean squared error (RMSE) is:

$$RMSE = \sqrt{\frac{SSD}{N}} \tag{7}$$

where N is the number of data points. Yet another measure is R^2 ("R-squared"), which compares the linear model to a "baseline" model which always predicts the same value (the mean of our observations: $\bar{y}$) for y, regardless of what x is.

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y}_i)^2} = 1 - \frac{SSD}{SST} \tag{8}$$

where SST is the "total sum of squares"

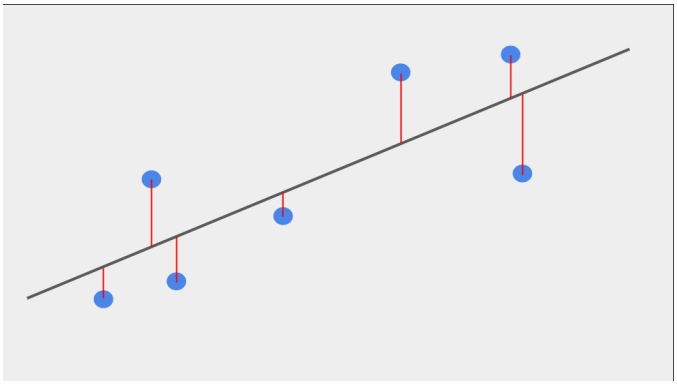

Figure 1: Adding the (squared) lengths of the red lines will tell us how good this fit is. Notice that the distances we sum are *not* the shorted lines from the point to the line, but rather the vertical distance to the line.

2.3 Exercise

Which of the equations below equals $y = ax + b$?

1. $$[y] = \begin{bmatrix} x \\ 1 \end{bmatrix} \begin{bmatrix} b \\ a \end{bmatrix}$$

2. $$[y] = [x \quad 1] \begin{bmatrix} a \\ b \end{bmatrix}$$

3. $$[y] = [x \quad 1] \begin{bmatrix} b \\ a \end{bmatrix}$$

4. $$[y] = \begin{bmatrix} x \\ 1 \end{bmatrix} [b \quad a]$$

2.4 Rewrite the problem using linear algebra

It might seem strange to do this now, but it will help us find a solution to the problem and help us use the technique when we have many input and/or output variables.

The linear function that does the prediction is:

$$\begin{bmatrix}\hat{y}\end{bmatrix} = \begin{bmatrix}x & 1\end{bmatrix} \begin{bmatrix}a \\ b\end{bmatrix} \tag{9}$$

To determine the parameters a and b, we need to consider all the pairs of data points at once. We have as many equations as we have data point pairs (N). First, let's stack the y_i in a vector.

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \tag{10}$$

Next, observe that we can write the vector of function predictions like this:

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \approx \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_N \end{bmatrix} = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \\ x_N & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} \tag{11}$$

Let's give names to the vectors and matrices:

$$\boldsymbol{y} \approx \hat{\boldsymbol{y}} = \boldsymbol{X}\boldsymbol{\beta} \tag{12}$$

Remember, at this point we have values for the x_i and y_i (the $\boldsymbol{X}$ matrix and the $\boldsymbol{y}$ vector). We need to find the best values for a and b (the $\boldsymbol{\beta}$ vector).

2.5 Finding the solution

We can use our linear algebra to find the solution. One way is using the normal equations:

$$\boldsymbol{\beta} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y} \tag{13}$$

2.6 Exercise

We can't use the inverse of $\boldsymbol{X}$ to solve for the best parameters. Three of the statements below give correct reasons for why, which one is wrong?

1. We might not be able to multiply $\boldsymbol{X}$ and $\boldsymbol{y}$ because their sizes are incompatible.

2. $\boldsymbol{X}$ might not be square, and non-square matrices don't have inverses.
3. There might not exist parameters ($\boldsymbol{\beta}$) that solve the equation $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta}$
4. We want to find the best approximation, and it might have non-zero error.

2.7 Derive the solution

where does the equation for the solution come from?

$$\begin{aligned} C &= (\boldsymbol{y} - \hat{\boldsymbol{y}})^T(\boldsymbol{y} - \hat{\boldsymbol{y}}) \\ &= (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) \\ &= \boldsymbol{y}^T\boldsymbol{y} - \boldsymbol{y}^T\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{y} + \boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\beta} \\ &= \boldsymbol{y}^T\boldsymbol{y} - 2\boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{y} + \boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\beta} \end{aligned} \tag{14}$$

Now take the derivative:

$$\begin{aligned} \frac{dC}{d\boldsymbol{\beta}} &= \frac{d}{d\boldsymbol{\beta}}(\boldsymbol{y}^T\boldsymbol{y} - 2\boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{y} + \boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\beta}) \\ &= (-2\boldsymbol{X}^T\boldsymbol{y} + 2\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\beta}) \end{aligned} \tag{15}$$

Set it equal to zero and solve for $\boldsymbol{\beta}$:

$$\begin{aligned} 0 &= (2\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\beta} - 2\boldsymbol{X}^T\boldsymbol{y}) \\ 0 &= 2(\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{X}^T\boldsymbol{y}) \\ 0 &= \boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{X}^T\boldsymbol{y} \\ \boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\beta} &= \boldsymbol{X}^T\boldsymbol{y} \\ \boldsymbol{\beta} &= (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y} \end{aligned} \tag{16}$$

2.8 SVD exercise

$$\boldsymbol{\Sigma} = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \end{bmatrix} \tag{17}$$

Which of the matrices below makes this true $\boldsymbol{\Sigma}\boldsymbol{X} = \boldsymbol{I}_{2\times 2}$

$$\boldsymbol{A} = \boldsymbol{X} = \begin{bmatrix} 0.25 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \end{bmatrix} \tag{18}$$

$$\boldsymbol{B} = \boldsymbol{X} = \begin{bmatrix} 0.25 & 0 \\ 0 & 10 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{19}$$

$$\boldsymbol{C} = \boldsymbol{X} = \begin{bmatrix} 0.25 & 0 \\ 0 & 10 \end{bmatrix} \tag{20}$$

2.9 Normal equations and the SVD

The matrix we get by solving the normal equations is called the pseudo-inverse:

$$\boldsymbol{X}^{\dagger} = (\boldsymbol{X}^T \boldsymbol{X})^{-1} \boldsymbol{X}^T \tag{21}$$

The singular value decomposition (SVD) of the matrix makes the pseudo-inverse *very* easy to compute. Remember the SVD is:

$$\boldsymbol{X} = \boldsymbol{U} \boldsymbol{\Sigma} \boldsymbol{V}^T \tag{22}$$

where $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthonormal matrices, and $\boldsymbol{\Sigma}$ is a non-square "diagonal" matrix.

$$\boldsymbol{X}^{\dagger} = \boldsymbol{V} \boldsymbol{\Sigma}^{\dagger} \boldsymbol{U}^T \tag{23}$$

where $\boldsymbol{\Sigma}^{\dagger}$ takes the reciprocal of the non-zero diagonal elements of $\boldsymbol{\Sigma}$, and transposes.

(Caveat: to be correct in general, all the transpose operations I wrote in this section should really be conjugate transpose operations).

3 Multi-variable linear regression

Now that we know how to write the linear regression problem as a matrix equation, it is relatively straightforward to extend it to the multi-variable case. In this set-up, we have several dependent variables to predict, and also have several independent variables to predict them from.

Suppose we want to predict two variables from two other variables. Say the two things we want to predict are:

1. y_1: the price of eggs in one month
2. y_2: the price of eggs in one year

and the two variables we can measure are:

1. x_1: the price of eggs today
2. x_2: the price of milk today

The equations will then look like:

$$\begin{aligned} \hat{y}_1 &= \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 \\ \hat{y}_2 &= \beta_0 + \beta_1 x_1 + \beta_2 x_2 \end{aligned} \tag{24}$$

We can write this system of equations with matrices like this:

$$\begin{bmatrix} \hat{y}_1 & \hat{y}_2 \end{bmatrix} = \begin{bmatrix} 1 & x_1 & x_2 \end{bmatrix} \begin{bmatrix} \alpha_0 & \beta_0 \\ \alpha_1 & \beta_1 \\ \alpha_2 & \beta_2 \end{bmatrix} \tag{25}$$

3.1 Exercise

Suppose we need to predict two output variables from three input variables, with an offset. What size will the matrix $\boldsymbol{X}$ be?

1. 3×2
2. 2×3
3. 4×3
4. 3×4

4 Polynomial regression

What equation does this correspond to?

$$\begin{bmatrix} \hat{y} \end{bmatrix} = \begin{bmatrix} 1 & x & x^2 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix} \tag{26}$$

Is this still linear regression? - yes! Notice that the function is non-linear in x, but it *is* linear in β_i!

Conclusion - we can fit polynomials with linear regression!

4.1 Exercise

Generalize this - write down matrix equation to fit a second-degree polynomial to two dependent variables (y_i) from three independent variables (x_i) with N datapoints.

There will be some second order terms (don't forget the "cross-terms" like $x_1 x_2$, which show up in multivariable polynomials), some first order terms, and the column of "ones."

The $\boldsymbol{X}$ matrix will have N rows, and how many columns?

- 3
- 5
- 10
- 13

5 Regress with basis functions

We can go even further than polynomials and use arbitrary functions of our independent variable(s), as long as the functions form a *basis*.

$$\begin{bmatrix}\hat{y}\end{bmatrix} = \begin{bmatrix}\phi_1(x) & \phi_2(x)\end{bmatrix} \begin{bmatrix}\beta_1 \\ \beta_2\end{bmatrix} \tag{27}$$

Below are some example of commonly chosen basis funcions.

5.1 Polynomial basis

$$f(x) = \sum_{i=0}^{\infty} \beta_i x^i \tag{28}$$

5.2 Fourier basis

$$f(x) = a_0 + \sum_{i=1}^{\infty} \alpha_i \cos(ix) + \beta_i \sin(ix) \tag{29}$$

5.3 Radial basis functions

The Gaussian "RBF", is a common choice:

$$f(x) = \sum_{i=1}^{N} e^{(x-x_i)^2} \tag{30}$$

but in general:

$$f(x) = \sum_{i=1}^{N} \phi(||\boldsymbol{x} - \boldsymbol{x}_i||) \tag{31}$$

These are "radial", because in multiple dimensions, the function ϕ acts only on the scalar radius, or distance to data points x_i.

5.4 Exercise - Linear combinations

Lets say we have two "basis" functions:

$$\begin{aligned} f_1(x) &= x^2 \\ f_2(x) &= \sin(x) \end{aligned} \tag{32}$$

These span some subset of the space of all possible functions. Which of these expressions is *not* a linear combination of f_1 and f_2?

$$\begin{aligned} a(x) &= 5x^2 \\ b(x) &= x^2 \sin(x) \\ c(x) &= 0 \\ d(x) &= 2\sin(x) + 0.1x^2 \end{aligned} \tag{33}$$

6 Two "fancy" optimization

6.1 Weighted least squares

In weighted least squares, we don't sum errors across all data points equally. Rather, we weight errors for some differently than others. This can be a good idea if your measurements have different levels of uncertainty, for example. These weights are fixed during optimization, and therefore have to be chosen ahead of time.

There is a good theoretical reason to choose your weights to be the inverse standard deviation of your measurements. So if we weight our errors

like this:

$$\boldsymbol{W}^{1/2} = \begin{bmatrix} \sigma_1^{-\frac{1}{2}} & 0 & \dots & 0 \\ 0 & \sigma_2^{-\frac{1}{2}} & \dots & 0 \\ \vdots & & \ddots & \\ 0 & \dots & 0 & \sigma_N^{-\frac{1}{2}} \end{bmatrix} \tag{34}$$

with my sincerest apologies for the nasty notation. Notice that $\boldsymbol{W}^{1/2}$ is a diagonal matrix. If we weight our errors like this:

$$E = \boldsymbol{W}^{1/2}(\boldsymbol{y} - \hat{\boldsymbol{y}}) \tag{35}$$

then our cost becomes:

$$\begin{aligned} C_W &= (\boldsymbol{W}^{1/2}(\boldsymbol{y} - \hat{\boldsymbol{y}}))^T \boldsymbol{W}^{1/2}(\boldsymbol{y} - \hat{\boldsymbol{y}}) \\ &= (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^T \boldsymbol{W}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) \end{aligned} \tag{36}$$

and the normal equations are:

$$\boldsymbol{\beta} = (\boldsymbol{X}^T \boldsymbol{W} \boldsymbol{X})^{-1} \boldsymbol{X}^T \boldsymbol{W} \boldsymbol{y} \tag{37}$$

where I wrote $\boldsymbol{W} = (\boldsymbol{W}^{1/2})^T (\boldsymbol{W}^{1/2})$, and the notational pain we suffered pays off, maybe.

6.2 Ridge regression / Tikhonov regularization

Sometimes, the parameters we estimate are very sensitive to small changes in the data we collect. We might be willing to make a tradeoff that gives us a lower "variance" (more stable/consistent) estimate at the cost of "bias". This could involve introducing some prior belief / preference on what we think the parameters should be. Maybe the most common is called ridge regression (in statistics) or Tikhonov regularization elsewhere.

All we do is modify the cost function from before like this:

$$C_R(\boldsymbol{\beta}) = \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|^2 + \lambda \|\boldsymbol{\beta}\|^2 \tag{38}$$

where $\|\boldsymbol{y}\|^2 = \boldsymbol{y}^T \boldsymbol{y}$ is the "norm" of a vector. The "hyperparameter" λ describes how "strong" the regularization is.

Let's get even fancier, Tikhonov regularization is written like this:

$$C_T(\boldsymbol{\beta}) = \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|^2 + \|\boldsymbol{\Gamma}\boldsymbol{\beta}\|^2 \tag{39}$$

We can still solve this closed form, with modified normal equations:

$$\boldsymbol{\beta} = (\boldsymbol{X}^T\boldsymbol{X} + \boldsymbol{\Gamma}^T\boldsymbol{\Gamma})^{-1}\boldsymbol{X}^T\boldsymbol{y} \tag{40}$$