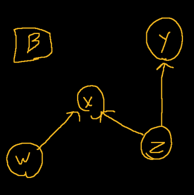


y: # of car accidents in NYC on May 6
x: # of umbrellas sold in NYC on May 6
z: rainfall in NYC on May 6
w: discounts on umbrellas on May 6

If x,y are correlated you can use x to predict y. If you're interested in prediction alone, you don't care about causation, you just need correlation.

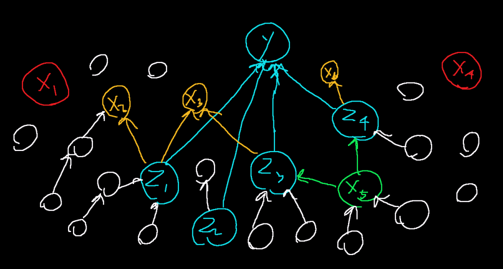
If we change x, does y change?

What is z? z is called a "lurking variable". So if you have only x, y and you don't observe z, z is "lurking". If we run a model with both x and z on y, will we see any correlation between x and y? Once the lurking variable is added to the model, there is nothing additional is learned through x.

$$\hat{y} = b_0 + b_x x + b_z z$$

b_x is the affect of x on yhat with z constant. We will come back to this interpretation soon.

What does a real data science context look like with x_1, ..., x_p?



You don't see this picture. Thus your x's can take on any of the three causal model scenarios: A, B and C. And you don't know their true connection to y! As p increases... what happens to the red x's? What do the red x's begin to look like? They're more likely to look real (yellow or green) because they become spuriously correlated.

Given this whole mess, what do OLS coefficients really mean? Not much at all... So let's try to interpret them carefully.

$$\hat{y} = g = b_0 + b_1 x_1 + b_2 x_2 + b_3 x_3, \quad b_1 = 3.26$$

Wrong interpretation of b_1 = 3.26 is:

"If I / you increases x_1 by one unit then y will increase by 3.26".

Wrong because (1) it's assuming x is causal (2) assuming true model is linear (3) not incorporating anything about the other x's.

More correct interpretation of b_1 = 3.26:

"When comparing two "mutually observed" observations (A) and (B) which are sampled in the same fashion as observations in D where (A) has an x_1 value one unit larger than the x_1 value of observation (B) but share the same values of the other x's then (A) is predicted to have a response that differs by b_1 units (+/- error in its estimation) in y from the predicted response of (B) on average assuming the linear model is true i.e.

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \mathcal{E}, \quad E[\mathcal{E}] = 0$$

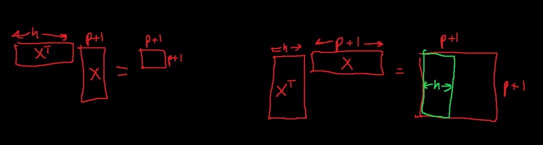
Note how this interpretation is not what you want. It is very weak. Too bad!

Ridge Regression

We're doing OLS so $\mathcal{H} = \{ \vec{w} \cdot \vec{x} : \vec{w} \in \mathbb{R}^{p+1} \}$

$$\vec{b} = \underbrace{(X^\top X)^{-1}}_{\text{must be invertible}} X^\top \vec{y}$$

If p+1 > n and X^T X is p+1 x p+1 and X^T X has at most rank n but here rank n is not full rank



$\Rightarrow X^\top X$ is $\underline{\text{not}}$ invertible

In the p+1 > n case, you're in trouble. You can't do OLS. What can we do? Can we use a trick to make X^T X invertible? Consider the following idea by Hoerl and Kennard in 1970:

$$\text{let } \vec{b}_{ridge} := (X^\top X + \lambda I_{p+1})^{-1} X^\top \vec{y} \quad \text{where } \lambda \neq 0 \text{ but } \lambda \approx 0$$

$$\Rightarrow X^\top X + \lambda I_{p+1} \approx X^\top X$$

This works as long as $X^\top X + \lambda I_{p+1}$ is invertible. Proof:

$X^\top X = V D V^{-1}$, V's columns are eigenvectors and D is a diagonal matrix with eigenvalues λ_j's for each of the eigenvectors in the corresponding entry. If X^T X is not full rank, then some of those diagonal entries in D will be 0's (the number of nonzero eigenvalues is the rank).

$$X^\top X + \lambda I = VDV^{-1} + \lambda I = VDV^{-1} + \lambda I V V^{-1}$$
$$= VDV^{-1} + V(\lambda I)V^{-1} = V(D + \lambda I)V^{-1}$$

$$D + \lambda I = \begin{bmatrix} \lambda_1 & & & & & 0 \\ & \lambda_2 & & & & \\ & & \ddots & & & \\ & & & \lambda_r & & \\ & & & & 0 & \\ 0 & & & & & \ddots \;\; 0 \end{bmatrix} + \begin{bmatrix} \lambda & & 0 \\ & \lambda & \\ & & \ddots \\ 0 & & & \lambda \end{bmatrix} = \begin{bmatrix} \lambda_1 + \lambda & & & & 0 \\ & \ddots & & & \\ & & \lambda_r + \lambda & & \\ & & & \lambda & \\ 0 & & & & \ddots \;\; \lambda \end{bmatrix}$$

D + λI has no zero entries on its diagonal. Hence:

$$(X^\top X + \lambda I)^{-1} = (V(D + \lambda I)V^{-1})^{-1} = V \underbrace{(D + \lambda I)^{-1}}_{\text{invertible}} V^{-1} \quad \checkmark$$

What does b_ridge correspond to? Remember b_OLS was the result of minimizing SSE over all possible w. Consider the following minimization problem:

$$A: \vec{b}_{ridge} = \underset{\vec{w} \in \mathbb{R}^{p+1}}{\text{argmin}} \{ SSE + \lambda \|\vec{w}\|^2 \} \Rightarrow w_j\text{'s closer to zero}$$

λ is called a regularization hyperparameter

$$(\vec{y} - X\vec{w})^\top (\vec{y} - X\vec{w}) + \lambda \vec{w}^\top \vec{w} = \vec{y}^\top \vec{y} - 2\vec{w}^\top X^\top \vec{y} + \vec{w}^\top X^\top X \vec{w} + \vec{w}^\top (\lambda I) \vec{w}$$
$$= \vec{y}^\top \vec{y} - 2\vec{w}^\top X^\top \vec{y} + \vec{w}^\top (X^\top X + \lambda I) \vec{w}$$

$$\frac{d}{d\vec{w}}[\quad] = -\cancel{2} X^\top \vec{y} + \cancel{2}(X^\top X + \lambda I)\vec{w} \overset{\text{set}}{=} \vec{0}_{p+1}$$

$$\Rightarrow X^\top \vec{y} = (X^\top X + \lambda I)\vec{w} \Rightarrow \vec{b}_{ridge} = (X^\top X + \lambda I)^{-1} X^\top \vec{y}.$$

FYI for those in the Bayesian class...

$$\vec{b}_{ridge} = \hat{\vec{\beta}}_{MAP} \text{ if } \vec{Y} | X \sim \mathcal{N}(X\vec{\beta}, \sigma^2 I) \text{ and } \beta_1, \ldots, \beta_{p+1} \overset{iid}{\sim} \mathcal{N}(0, \tau^2)$$
$$\Rightarrow \lambda = \frac{\sigma^2}{\tau^2}$$

Practically speaking, how do we pick λ? Model selection algorithm.

Ridge estimation induces bias but it reduces variance hopefully for a holistic reduction in MSE.

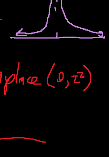
$$\|\vec{w}\|_1 := \sum_{j=1}^{p+1} |w_j|$$

Consider the following related algorithm:

$$\vec{b}_{Lasso} := \underset{\vec{w} \in \mathbb{R}^{p+1}}{\text{argmin}} \{ SSE + \lambda \|\vec{w}\|_1 \}$$

This has no closed form solution so you need to use a computer.

The solution is very harsh i.e. it will call a lot of the b_j's = 0. This allows for "feature selection" which is given a large set of possible features, it give you back a subset of "useful" features. It's an important "prestep" to predictive modeling we never really spoke about too much. It can be called the "Occams Razor-izer".



$$\vec{b}_{Lasso} = \hat{\vec{\beta}}_{MAP} \text{ for } \vec{Y} | X \sim \mathcal{N}(X\vec{\beta}, \sigma^2 I) \text{ but } \beta_1, \ldots, \beta_{p+1} \overset{iid}{\sim} \text{Laplace}(0, \tau^2)$$
$$\text{where } \lambda = \sigma^2 / \tau^2$$

Consider the following algorithm:

$$\vec{b}_{elastic} := \underset{\vec{w} \in \mathbb{R}^{p+1}}{\text{argmin}} \{ SSE + \lambda (\alpha \|\vec{w}\|_1 + (1 - \alpha) \|\vec{w}\|_2^2) \}$$

The elastic net algorithm combines both the ridge and lasso regularization terms using a linear combination hence introducing another hyperparameter α.

Midterm II ↑

EXTRA ↓