REGULARIZATION INTRODUCTION —

- Overfitting is a constraint in training data, where model overfit in training data and performs poorly in testing data.
- In general, regularization means to make things regular or acceptable. In ML, regularization is the process which regularize or shrinks the coefficient toward zero. In simple word, regularization discourage learning a more complex or flexible model, to prevent overfit.

How to address Overfitting?

- One way could be to reduce number of features in the model.
- But there will be loss of information due to discarding. Model will not have the benefit of all the information.
- So regularization can be used, where it will keep all the variables but also reducing the magnitude / altitude of features available.
- Thus regularization provide trade-off between accuracy & generalizability of model.

Type of Regularization —

i) Lasso Regularization (L1)
ii) Ridge Regularization (L2)
iii) Elastic Net Regularization.

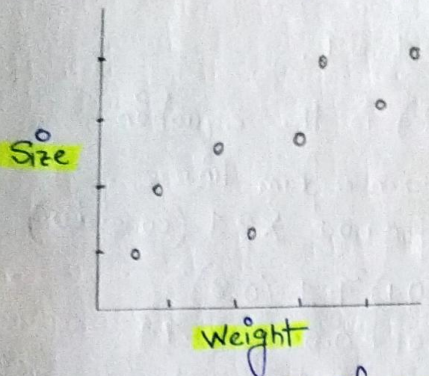
When do we need Regularization at first place?

- Suppose we load all the variables in the model and observe the performance of model. To find variable which are significant to be included and which are not, Regularization become handy to identify the variables that should remain in the model.

REGULARIZATION

RIDGE REGRESSION (L2 Regularization)

Let's take the example of weight and size measurement from bunch of mice.

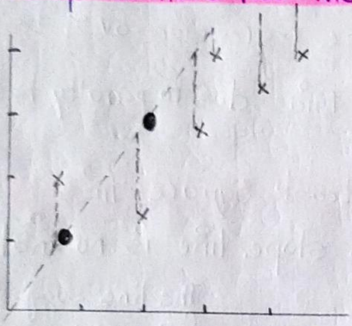


- Since these data looks relatively linear, we will use Linear Regression, aka least square to model the relationship between weight & size.
- So we will fit a line to data using Least Squares. In other word, we will find the line that results in the minimum sum of square residuals.

Suppose eq^n comes, $\boxed{size = 0.97 + 0.75(weight)}$

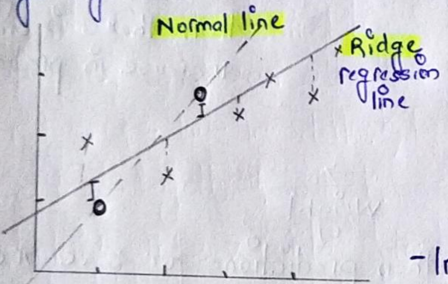
- When we have lot of measurements, we can be fairly confident that the Least Squares line accurately reflects relationship between size & weight.

What if we have two measurements?



- Since the line overlaps two data points, the minimum sum of squared residuals = 0.
- Suppose O → training set   x → testing set
- In training set, SSR = 0 but in test set, SSR is high.
- This means lines is high variance, in other word line is overfit in training dataset.

So ridge regression comes into the picture.



- Main idea behind Ridge Regression is to find a New line that doesn't fit the training data as well.
- In other word, we introduce a small amount of Bias into how the New line fit to the data.
- In return when we plot as testing set, we get small amount of Bias as well as Variance

In other word, by starting with worse fit, Ridge regression provide better result in test.

So how Ridge Regression works?

- When least squares values works, size = y-axis intercept + slope (weight)
  it minimizes sum of squared residuals.

In contrast, ridge regression works, size = y-axis intercept + slope (weight)
it minimize, sum of square residuals + λ * the slope^2
here the slope^2, adds a penalty to traditional least sq. method
λ, determine how severe penalty is

So lets calculate, ⇒ sum of square residuals + λ (slope)^2

For Normal least square, size = 0.9 + 1.3 (weight) is the equation.

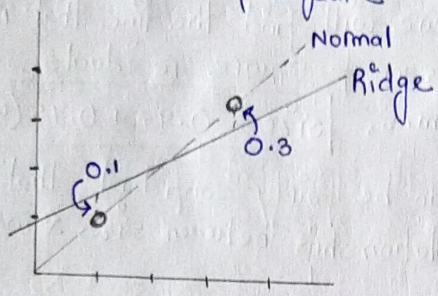
here, sum of square = 0, & slope = 1.3, for now $\lambda = 1$ (consider)

(because it pass through 2 points)

value = $0 + 1(1.3)^2 = 0 + 1.69 = 1.69$.

For Ridge regression, size = 0.9 + 0.8 (weight) is the equation.

here sum of square = $(0.3)^2 + (0.1)^2$, difference from line



slope = 0.8, for now $\lambda = 1$ (consider)

value = $(0.3)^2 + (0.1)^2 + 1\,(0.8)^2$

$= 0.09 + 0.01 + 0.64 = 0.74$.

So for Normal line, penalty = 1.69.

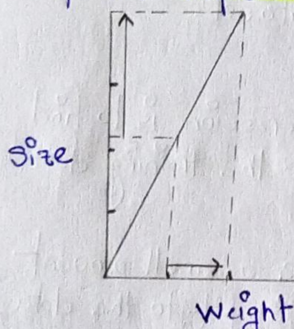
For ridge regression, penalty = 0.74.

Thus if we wanted to minimize sum of squared residuals plus the ridge regression penalty, we will choose ridge regression over Least Square line.

In short, Ridge regression line, which has small Bias due to penalty has less Variance.

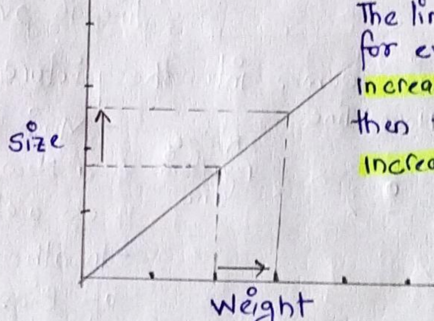
Why Ridge regression is working better than Least Square line?

If the slope line is steeper.



For every one unit increase in weight then prediction for size increase over two units.

If the slope line is normal.



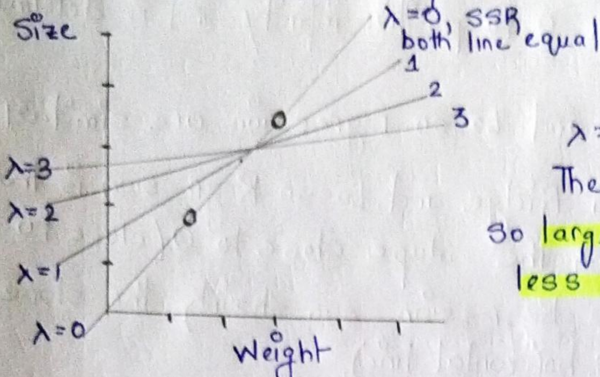
The line suggests that for every one unit increase in Weight then there is one unit increase in predicted size

In other word, when the slope of line is small, then predictions for size are much less sensitive to changes in Weight.

So in case of normal vs Ridge, Least Square line is much more steeper than the ridge line. Ridge regression penalty resulted in a line that has a smaller slope which means that predictions made with Ridge regression line are less sensitive to Weight than the Least Square Line.

So Now, lets see Ridge Regression tries to minimize, SSR + $\lambda(\text{slope})^2$

- λ can be any value from 0 to positive infinity.
when $\lambda = 0$, penalty = $\lambda(\text{Slope})^2 = 0(\text{Slope})^2 = 0$ is also zero.
So in that case SSR is there, therefore Ridge regression line = Least square line.



As we increase the λ from 0 to 1, slope become steeper.
$\lambda = 2$, more steeper and so on.
The larger we make λ, slope get horizontal
So larger λ gets, our prediction for size become less and less sensitive to Weight

- So how to decide what value to give λ?

- Use bunch of values for λ and use cross validation to determine which one result in the lowest variance.
- Ridge regression works with both Continuous Vs Continuous data like Size vs weight as well as Continuous Vs Categorical data like Size vs Diet (High, Low).
- In short, Ridge regression helps reduce Variance by shrinking parameters and making our prediction less sensitive to them.
  - If we have more than 2 parameters.
  In general, the Ridge Regression penalty contains all of the parameters except for the y-intercept.
- Ridge regression is used when we have parameter/feature > data points
  (> greater than)

Summary →
- When the sample sizes are relatively small then Ridge Regression can improve predictions made from new data (i.e, reduce Variance) by making the prediction less sensitive to training data. This is done by adding the Ridge Regression penalty to the thing that must be minimized.

$$\boxed{= \text{the sum of the squared residuals} + \lambda * \text{Slope}^2}$$

Ridge regression penalty itself is λ times sum of all squared parameters except y intercept & λ is determined using Cross Validation.

LASSO REGRESSION (L1 Regularization)

According to $\boxed{\text{Ridge regression} = \text{sum of square residuals} + \lambda * \text{slope}^2}$

If instead of slope^2, we use absolute slope, |slope| then it is

$\boxed{\text{Lasso regression} = \text{sum of square residual} + \lambda * |\text{Slope}|}$

Here also, λ can be any value from 0 to positive infinity and is determined using cross validation.

So most of the remaining thing, Ride and Lasso regression are similar.

Difference → Big difference between Ridge and Lasso Regression is that Ridge Regression can shrink the slope close to 0 (close to horizontal line) while Lasso Regression can shrink the slope all the way to 0 (perfect horizontal line).

So in ridge regression, variable value can shrink to very small but cannot be zero.

In lasso regression, variable value can be shrink equal to 0.

Example

Suppose, salary = slope + β_1 (experience) + β_2 (education) + β_3 (name of person) + β_4 (weight of person)

here we have to predict salary, 2 good variable → experience, education
2 useless variable → name, weight

In ridge regression, value will shrink but never $\neq 0$ for useless variable.
so all 4 variable will remain regardless of useless.

In lasso regression, values will shrink and useless variable will become 0.
so only 2 variables (useful) will remain.

Since, lasso regression can exclude useless variable from equations, it is little better than Ridge regression at reducing Variance in a model that contains lot of useless variables.

In contrast, Ridge Regression tends to do better (little) when most variable are useful.

Summary –

Ridge regression square the variable = $SSR + \lambda * \text{Slope}^2$

Lasso regression takes the absolute value = $SSR + \lambda * |\text{Slope}|$

But big difference is Lasso Regression can exclude useless variables from equation

This equation is simpler and easier to interpret.

ELASTIC NET REGRESSION (Type of Regularization)

- When a dataset include millions of parameters and when we have millions of parameters, we almost need to use some sort of regularization to estimate them.

However, the variables in those model might be useful or useless, we don't know in advance.

So how to choose Lasso (~~all~~ some variables are useful) or Ridge (all variables are useful)

There use Elastic Net Regression.

Elastic Net regression combines strength of Lasso and Ridge regression.

$$\boxed{= \text{sum of square residuals} + \lambda_1 * |\text{variable}_1| \ldots + |\text{variable}_n| + \lambda_2 * (\text{var}_1)^2 + \ldots + (\text{var}_n)^2}$$

λ_1 is for lasso regression penalty, λ_2 is for ridge regression penalty

We use Cross Validation on different combination of λ_1 and λ_2 to find best values.

If λ_1 and $\lambda_2 = 0$, then we get original least Square Parameters

$\lambda_1 \neq 0$ and $\lambda_2 = 0$, then we get lasso regression.

$\lambda_1 = 0$ and $\lambda_2 \neq 0$, then we get ridge regression.

$\lambda_1 \neq 0$ and $\lambda_2 \neq 0$, then we get elastic net regression.

Hybrid Elastic-Net Regression is especially good at dealing with situation when there are correlation between parameters. Because lasso tends to pick just one correlated term's and ~~es~~ eliminates others ~~but~~ and ridge tends to shrink all of the parameters for the correlated variables together.

By combining Lasso & Ridge, elastic-net groups and shrinks the parameters associated with correlated variables & leave them in equation or removes them all at once.

Summary -

Elastic Net regression, $\boxed{= \text{sum of square residuals} + \underbrace{\lambda_1 * |\text{var}_1| \ldots |\text{var}_n|}_{\text{lasso penalty}} + \underbrace{\lambda_2 * (\text{var}_1)^2 + \ldots + (\text{var}_n)^2}_{\text{Ridge penalty}}}$

It get the best out of two and also good at dealing with correlated ~~variables~~ parameters.