Lesson 7

Finding the Best Fit Model

Forward selection:

- Create models for each variable which only includes that single variable
- Calculate RSS and R2 for each model.
- Find the best model i.e. smallest RSS and largest R2
- Then combine the model for that best variable with each of the other variables to create models with two variables.
- Again, calculate RSS and R2 for each model.
- Find the best model
- Repeat
- Find which of the models from each number of variables gives you the best BIC, IAC or adjusted R2

In this selection method, you might miss a model which gives you the best score because you have limited your variable combinations from the beginning.

Backward selection:

- As above, but backwards i.e. start with a model which includes all variables

In order to be able to perform backward selection, we need to be in a situation where we have more observations than variables because we can do least squares regression when n is greater than p. If p is greater than n, we cannot fit a least squares model.

Mixed selection:

- Mixture of above

More info: https://gerardnico.com/wiki/data_mining/stepwise_regression

R2 (R-squared) is the metric for linear regression. It represents the proportion of the variance in the dependent variable that is predictable from the independent variables i.e. the higher R-squared is, the better your model.

In linear models, residual error must be normal with median close to zero.

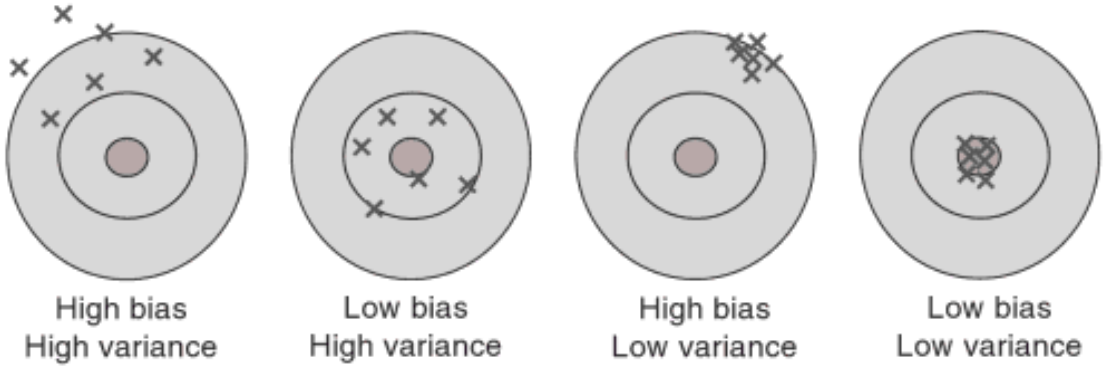
However, we need a metric which summarises the error in our model in one value. We use the mean squared error for this.

Mean Squared Error:

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2$$

Where y is each target value and y-hat is the predicted value by the model. N is the number of points.

OLS: Ordinary Least Squares (linear regression method)
For a given matrix, X, solve for the least amount of square error for y (observations - predictions ^2) (assuming that X is unbiased and representative of the population).

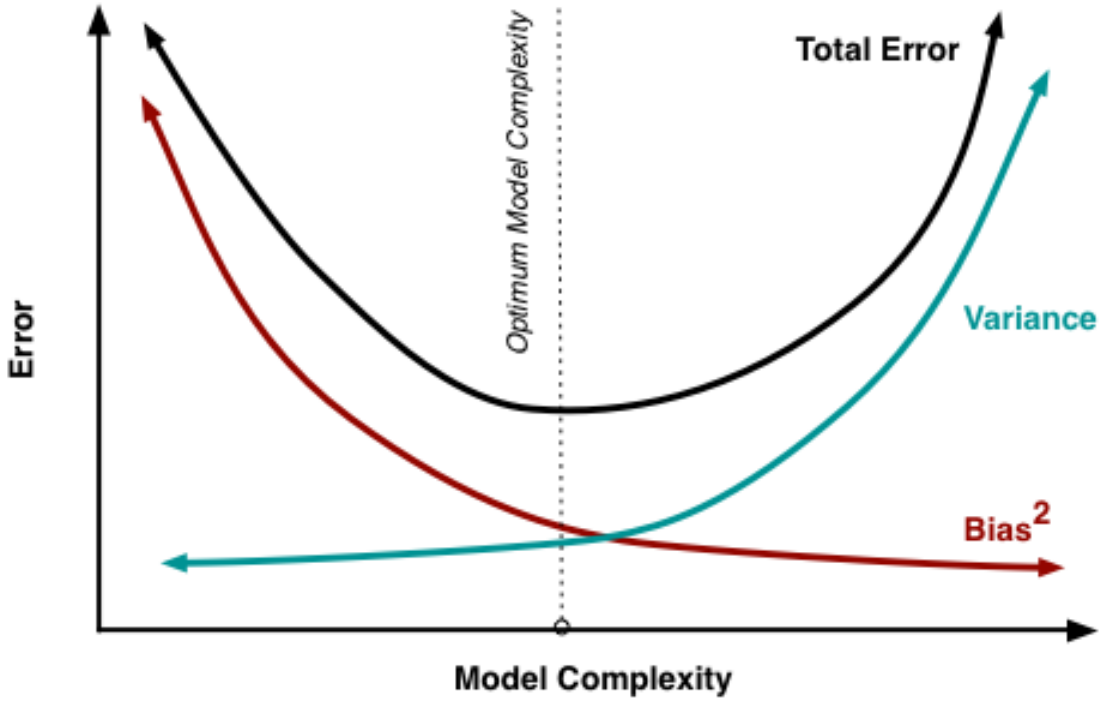


When our error is biased, it means the model's prediction is consistently far away from the actual value. This could be a sign of poor sampling and poor data.

One objective of a biased model is to trade bias error for generalised error. We prefer the error to be more evenly distributed across the model. This is called error due to variance.

The mean squared error measures bias.

We want our model to generalise to data it hasn't seen even if doesn't perform as well on data it has already seen.



Cross validation:

- Cross validation can help account for bias
- Cross validation is a method by which you separate the data into different cross sections, generate a model for each cross section, measure the performance of each and then take the mean performance
- This method swaps bias error for generalised error

K-fold cross validation:

- Split the data into k groups
- Train the model on all segments except 1
- Test the model performance on the remaining set

Regularisation:

- Regularisation is an approach to building models which protects against overfitting (i.e. fitting the training dataset so well that it becomes biased and overconfident)
- Regularisation becomes an additional weight to coefficients, shrinking them closer to zero
- Lasso regression (L1) adds the extra weight to coefficients
- Ridge regression (L2) adds the square of the extra weight to coefficients
- Use Lasso when you have more features than observations and Ridge otherwise

Lasso (L1) and Ridge (L2) regression: https://www.analyticsvidhya.com/blog/2017/06/a-comprehensive-guide-for-linear-ridge-and-lasso-regression/

Grid search is a method which searches through all the models with the given

parameters (depending on which model method you are using) to find the best solution.

Reminder:

- Training set: data set which you use to build the model
- Testing set: data set which you use to test the model i.e. you use this model to predict the outcomes

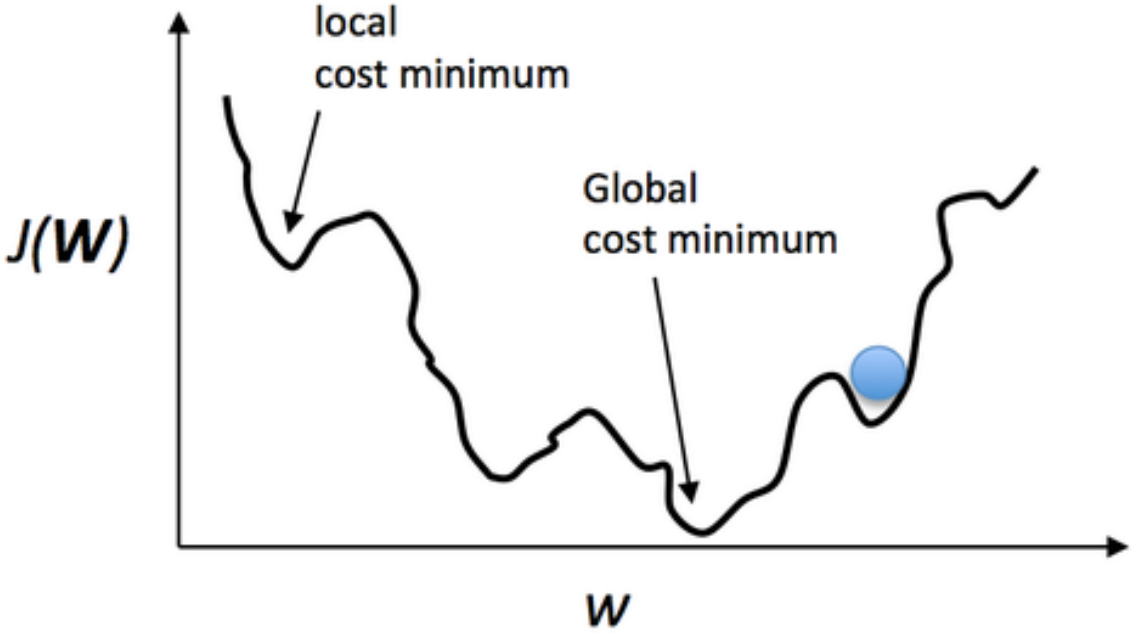
Gradient descent can also help us to minimise error.

How gradient descent works:

- A random linear solution is provided as a starting point
- The solver attempts to find a next "step": take a step in any direction and measure the performance. (Moves along the cost function).
- If the solver finds a better solution (i.e. lower MSE), this is the new starting point.
- Repeat these steps until the performance is optimised and no "next steps" perform better. The size of steps will shrink over time.

This graph shows the logistic cost function for a one-variable linear model.
Variable = W
Cost function = J(W)
Learning Rate = the rate at which you move W when trying to find the minimum

You could end up reaching a local cost minimum instead of the global cost minimum.



Source: https://sebastianraschka.com/faq/docs/logisticregr-neuralnet.html

Gradient descent works best when:

- We are working with a large dataset. Smaller datasets are more prone to error.
- Data is cleaned up and normalised.

Gradient descent is significantly faster than OLS. This becomes important as data gets bigger.

Gradient Descent can be tuned with:

- The learning rate: how aggressively we solve the problem
- Epsilon: at what point do we say the error margin is acceptable
- Iterations: when should be we stop no matter what