DDCS Report Joel Bearn 1802334

Finding functions

For linear regression the following equations are needed:

Sum Squared Error: $SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$

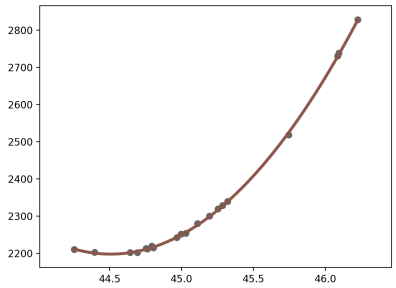
Estimated Weights: $\hat{W} = (X^T X)^{-1} X^T Y$

Finding the Polynomial order

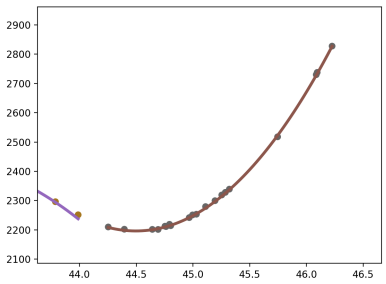
The estimated weight equation for linear regression can be expanded such that any number of parameters can be added including polynomials. I decided that the polynomial order that is the best model for the data is order 3 (i.e including an X cubed term). To reach this conclusion I initially looked at plots of different orders and then performed K-fold cross validation to confirm my findings.

Plots of different orders:

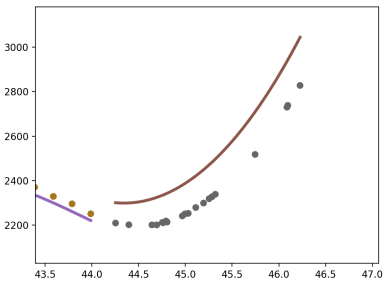
Adv_3 segment 6



Order 2



Order 3



Order 4

Order 2 plots did seemed to be good fits by eye, as did plots of order 3. Plots of order 4 started to deviate from the data.

Cross Validation to confirm polynomial order

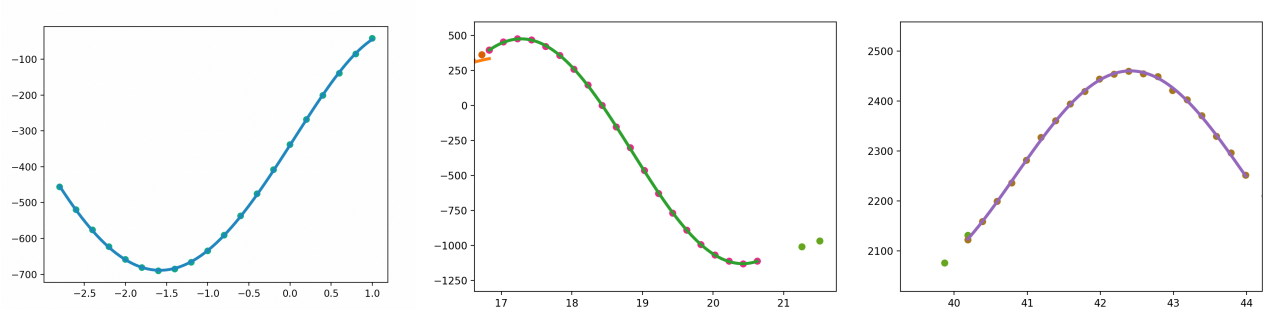
I used a K-fold cross validation to confirm the order of the polynomial. This was to combat overfitting which is where the estimated model fits the sample data too tightly and is not an accurate predictor of new values. Overfitted polynomials with too high an order would fit the data itself very well but when run on cross validation would fail as it was fitting the training data points too tightly so when the test data was re-added the error was significantly higher than the model with the correct order.

A couple of the segments estimated an order of 2. This means for those segments the X cubed term must have been very minimal. The overwhelming majority of polynomial segments estimated an order of 3.

Finding the Unknown Function

Looking at the segments that were most likely the unknown function (such as basic_5 and adv_3 segments 3 and 5) I could see there was a strong sinusoidal relationship. As these sinusoids did not fluctuate around y=0 I added a bias column to my matrix. Having tried both cosine and sine functions I could see that the sine function fitted remarkably well and the cosine functions did not. I did not add a linear parameter to the function as in the context of the coursework being during the Cold war it would make sense to have a sine signal of a single frequency and amplitude.

Plots showing that Sine(x)+c is a suitable guess for the unknown function:



Running these segments through cross validation the error was found to be very small so I am fairly confident this is the correct function.

Determining the correct function for a segment

I went about selecting the best model for each segment by using k-fold cross validation. Cross validation is a good method to avoid overfitting. If the model fits the training data too tightly then the predictions for the test data will result in larger errors from the true value than a model which is more likely the correct function.

My method was specifically ‘K-fold’ cross validation. In my method the data points of a segment are divided into K bins. Cross validation is then performed K times, each time the nth bin is excluded and used as the test data and the remaining points are used as training data. An average of the sum squared error is then found. The model which provided the lowest average error was selected to be the correct function.

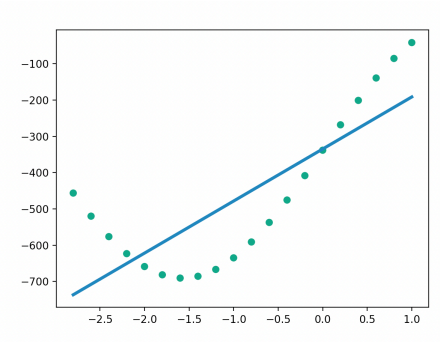
Initially I used single cross validation which chose n random points as test data. Then whichever of the models (linear, polynomial, unknown) gave the smallest sum square error was picked to be the correct function. The advantage of a single cross validation is that it is quick in comparison to K-fold validation as the validation is only done once. The disadvantage is that a different random seed may result in different models being picked. In the context of a nuclear signal this is quite a

significant disadvantage as false positives could cause unnecessary panic and false negatives could have deadly consequences.

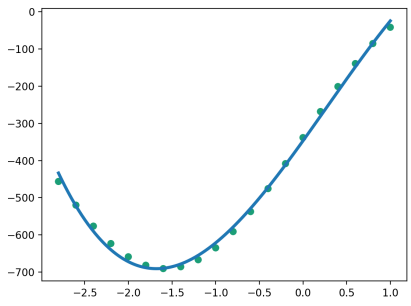
Therefore I chose K-fold cross validation as it does not give different answers depending on a random seed. I thought this was a valid trade off for the extra computation K-fold cross validation takes.

There was an instance where my model selected polynomial instead of linear to be the correct model. This is acceptable as the polynomial model produced was still a good predictor for the data and is almost linear rather than being an overfitted model. Each segment only has 20 data points so overfitting is more likely to occur than if there was a more substantial number of points per segment.

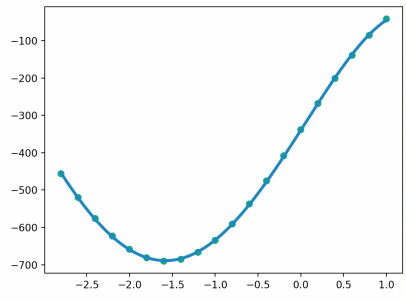
Example 1: Determining the correct function (Sine) for basic_5



Linear



Polynomial

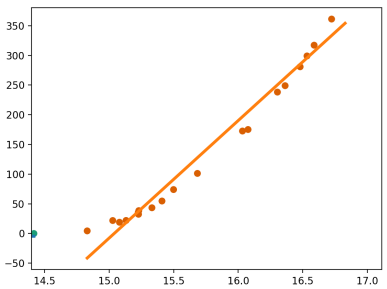


Sine

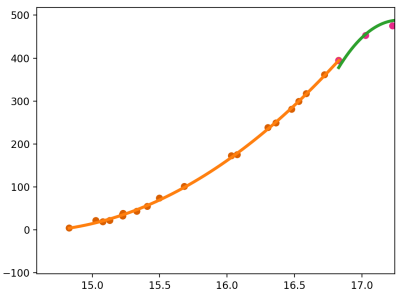
```
basic_5.csv
K FOLD CROSSVALIDATION ERR  linear 409840.70662135934
K FOLD CROSSVALIDATION ERR  poly 40876.40611487372
K FOLD CROSSVALIDATION ERR  unknown 1.474223259677027e-26
Chose UNKNOWN
```

The data shows that the average cross validation error was very large for a linear model, reasonably large for a polynomial model and very small for the sine model. Therefore my algorithm chose sine to be the best model. The plots above confirm this was the correct decision as the linear plot is a very poor estimate, the polynomial plot is fairly good but the sine plot fits near perfectly.

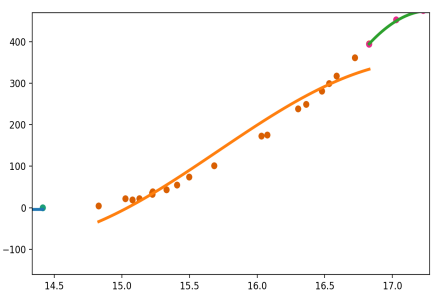
Example 2: Determining the correct function (Polynomial) for Adv_3 segment 2



Linear



Polynomial

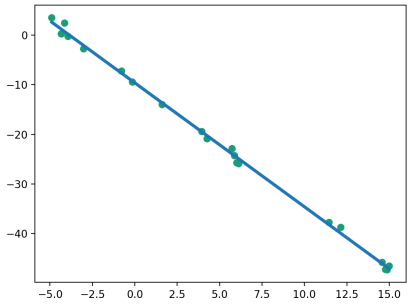


Sine

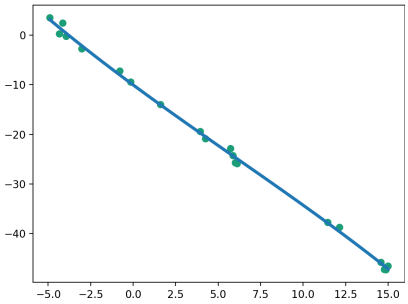
```
K FOLD CROSSVALIDATION ERR  linear 8473.130107689354
K FOLD CROSSVALIDATION ERR  poly 153.0091437837537
K FOLD CROSSVALIDATION ERR  unknown 11517.661958487948
Chose POLY
```

It is clear to see from the plots that the polynomial function is the best predictor and my cross validation errors confirm this as the polynomial error is more than an order of magnitude smaller than the linear and unknown errors.

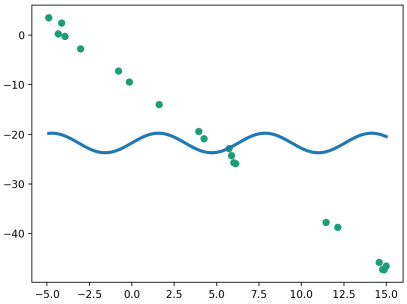
Example 3: Determining the correct function (Linear) for noise_2 segment 3



Linear



Polynomial



Sine

```
noise_1.csv
K FOLD CROSSVALIDATION ERR  linear 3.3915825209218093
K FOLD CROSSVALIDATION ERR  poly 18.010877153774125
K FOLD CROSSVALIDATION ERR  unknown 4105.071200265554
Chose LINEAR
```

As Linear functions are a subset of polynomial functions it is not a surprise that the linear and polynomial plots look very similar as the polynomial has very small X^2 and X^3 terms. Segments like these are why I chose K-fold cross validation as on single cross validation the random seed may have often resulted in the polynomial being chosen.

Conclusion

To conclude my code accurately predicts the correct function type. It is especially good at determining which segments are the unknown / sine function. In the context of an unknown signal in 1983 this would give the operator of the nuclear early warning device confidence that the signal received was in fact the unknown function / sine wave and therefore early warning systems could be deployed.