Regression and Model Selection

Prediction with Least-squares

Given a set of observed inputs X=(X1, X2, … Xp), we wish to devise a model that predicts the true value Y of the output variable. As an example, X can be the words in an email, and Y the indication if that email is spam Y>0 or not Y<0.

A model can only make a prediction $\hat{Y}$ that is the best guess of the true value Y. Thus, the model can be formalized as a linear model:

$$\hat{Y} = \beta_0 + \sum_{j=1}^{p} \beta_j \cdot X_j$$

The goal is then to learn the model parameters $\beta = (\beta_0, \dots, \beta_p)$. The least-squares approach is the most popular approach to learn these parameters. The goal is to minimize the residual sum of squares between the true label and the predicted label:

$$RSS(\beta) = \sum_{i=1}^{N} (\hat{y_i} - y_i)^2$$

The Residual Sum of Squares gives an indication of how well a model the model can make a prediction.

Model fitting

Consider the simple case of a polynomial fitting with the following data

x	y
0.5588	1.1561
0.5475	1.1408
0.1899	0.5806
0.3065	0.7987
…	…

$$y = x^p \beta_p + x^{p-1}\beta_{p-1} + \cdots + x^2 \beta_2 + x^1 \beta_1 + \beta_0$$

In this case we need to compute the polynomial coefficients $\beta = (\beta_0, \dots, \beta_p)$. The following code is an example of how to fit a third-degree polynomial to the data:

```
sinpoints=importdata('sin_mat.dat');
dataSet=sinpoints;
numSets=200
numPoints=size(dataSet,1);
coefs=polyfit(dataSet(:,1),dataSet(:,2),3);
```

Discuss how you can cast the polynomial fitting problem as a least-squares problem. Consider that the X matrix can be written as $X = (x^p, x^{p-1}, \ldots, x^2, x^1, 1)$.

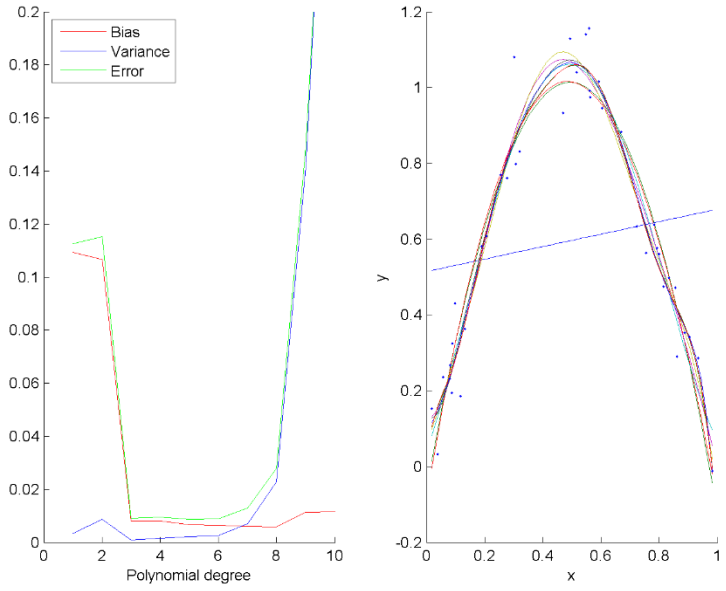
Given the polynomial coefficients, we can compute the predicted values and the Residual Sum of Squares:

```
predicted=polyval(coefs,dataSet(:,1));
rss =mean((dataSet(:,2)-predicted).^2);
```

To examine the observed data versus the computed model, we can plot this information:

```
xs=linspace(min(dataSet(:,1)), max(dataSet(:,1)), 50)
ys=polyval(coefs,xs);
plot(xs,ys,'-r');
hold on
plot(dataSet(:,1),dataSet(:,2),'xb')
```

Repeat the same procedure for different polynomial degrees. Plot the RSS for each polynomial degree.



The main questions arising from these graphs are:

- Which polynomial degree should we chose? (model complexity)
- How can we examine the model behavior as the complexity increases?
- How to estimate the error on the test set?

Exercise 1: Bias-variance decomposition with bootstrapping

Download the file sin.dat (or sin_mat.dat for MATLAB) to your working folder. Load it with the importdata command. This will create a 40x2 matrix containing a simulated dataset of a sinusoidal curve (x on the first column, y on the second).

Using bootstrapping, with S=10 sets, plot the estimated bias2 and variance values as a function of the polynomial degree used to fit these data, ranging from degree 0 (a constant) to degree 10. The error of a model is decomposed as follows:

$$Err = E\left[(Y - \hat{Y})^2 | X\right]$$

$$Err = \sigma_\varepsilon^2 + \left[E[\hat{Y}] - Y\right]^2 + E\left[(E[\hat{Y}] - \hat{Y})^2\right]$$

$$Err = Irreducible\ Error + Bias^2 + Variance$$

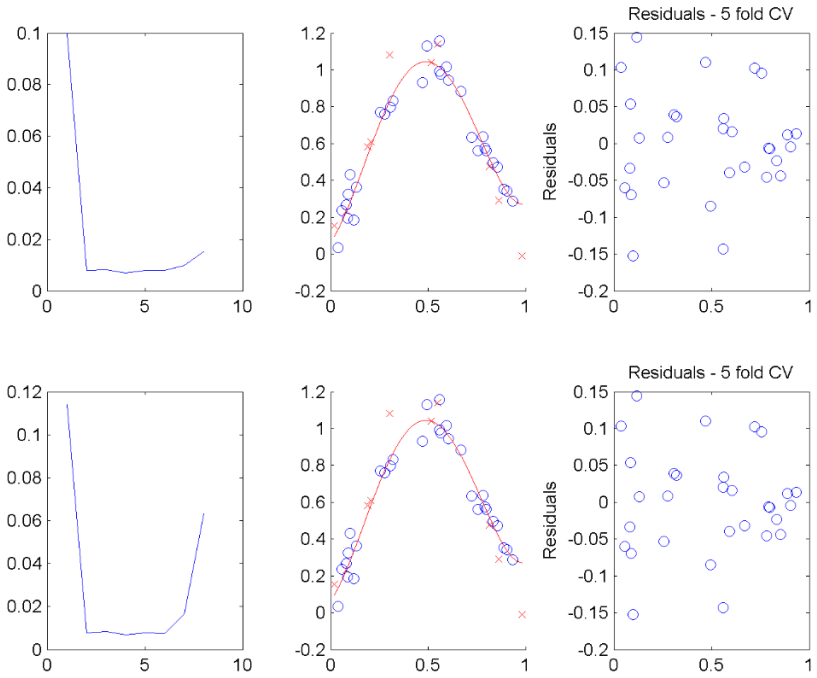
$$Bias^2 = \frac{1}{N}\sum_{i=1}^{N}\left(\left(\frac{1}{S}\sum_{s=1}^{S}\hat{y}_i^s\right) - y_i\right)^2 \qquad Variance = \frac{1}{S}\sum_{s=1}^{S}\left(\frac{1}{N}\sum_{i=1}^{N}\left(\left(\frac{1}{S}\sum_{s=1}^{S}\hat{y}_i^s\right) - \hat{y}_i^s\right)^2\right)$$

Create the figure shown on the previous page.

Exercise 2: Model selection with Cross-Validation.

From the same data as in exercise 1, extract 10 points for a test set. Use the remaining points as a training set for cross-validation:

1. Using 5-fold cross-validation, select the appropriate degree for your polynomial (assume from degree 1 through 8). Then train your selected model with the training set and estimate the generalization error with the test points.
2. Repeat the procedure with leave-one-out cross-validation.
3. Compare both procedure by plotting the CV-error, the polynomial with the selected degree, and residuals to make a visual evaluation of the resulting fit.

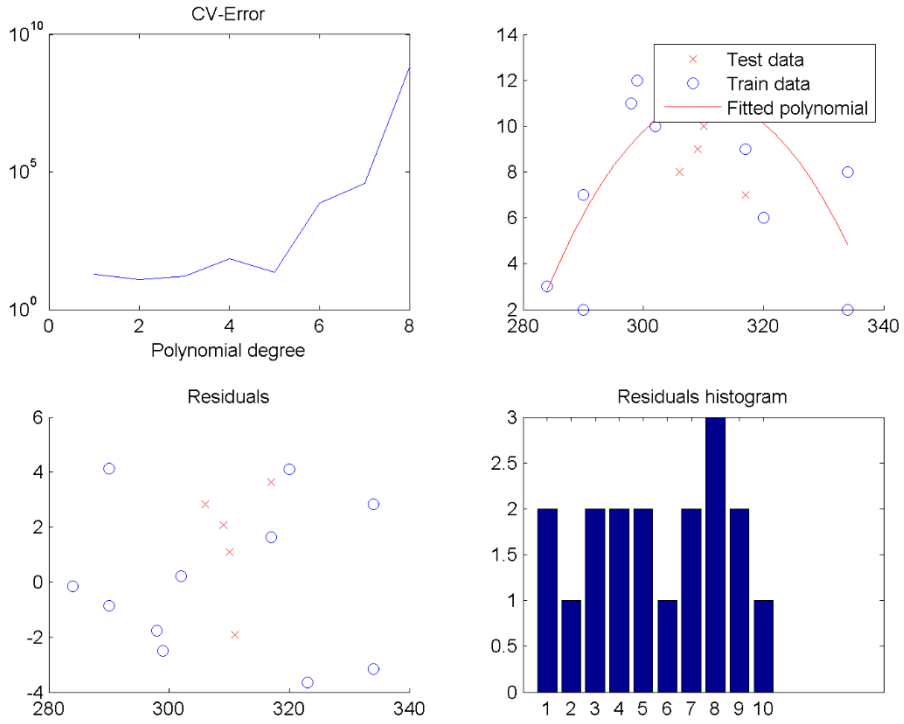


Exercise 3: Turtles and eggs

Load the turtles_mat.dat file. Each of these 18 points (one per row) represents the carapace length (mm) and number of eggs on a set of 18 gopher turtles. The goal is to predict the eggs size of a turtle given its carapace length.

Assume that the model is a polynomial and use CV to find the best fit. Retain 6 points for testing and use cross-validation to find the best polynomial fit (up to degree 8) for these data. Note that since there are very few data points for training, use a leave-one-out cross-validation scheme.

Estimate the error on the test set and plot the residuals to evaluate the fit graphically.



The turtle dataset was copied from the Handbook of Biological Statistics, and originally from Ashton, K.G., R.L. Burke, and J.N. Layne. 2007. Geographic variation in body and clutch size of gopher tortoises. Copeia 2007: 355-363.