MCS507 Project Two:
Study a Mathematical Software Package: Earth

Prepared By: Adam McElhinney October 28, 2012

---

1 Assignment One: Overview and Illustrative Example.

Our first objective is to identify the main problem this software aims to solve. We then provide an overview of the software package and the algorithms used. We conclude with an illustrative example that displays the utility of this software.

1.1 Main Problem this Software Aims to Solve

The Earth software package is an implementation of the Multivariate Adaptive Regression Splines (MARS) technique developed by Jerome H. Friedman of Stanford University. Interestingly, the reference MARS is copywritten, thus open source implementations of the software are referred to as Earth. The purpose of this technique is similar to that of all regression techniques, namely to fit a function to a set of data. However, the Earth software specifically seeks to do this in a manner that has the following advantages:

1. Ability to easily handle non-linear data sets
2. Ease of interpretation
3. Ability to handle large datasets
4. Automatic variable selection

1.2 Overview of the Software and Algorithms

The Earth software package relies on the application of splines to construct the function to predict the target variable. Broadly speaking, a spline is simply a function constructed in various segments from other polynomial functions. The MARS technique relies on a type of splines known as hinge functions, which can be represented in the form:

$$h_j = \max(0, x - c) \tag{1}$$

The constant c is referred to as the knot. The Earth software then uses these functions to divide the data into mutually exclusive segments and then fit a model to each individual function. This is done in such a manner as to minimize some objective function. One commonly used objective function is the residual sum of squares (RSS).

$$RSS = \sum_{i=1}^{n} (y_i - f(x_i))^2 \tag{2}$$

To illustrate, let us consider the typical regression equation where we seek to build a model that predicts the value of the y_i-th obvseration based on some values of x.

$$y_i = \beta_0 + \sum_{i=1}^{M} \beta_i * x_i \tag{3}$$

The Earth software then modifies this model to have the following form (where h_i represents the hinge function in equation 1).

$$y_i = \beta_0 + \sum_{i=1}^{M} \beta_i * h_i(x_i) \tag{4}$$

To fit this model, the Earth software relies on two distinct steps.

1. Forward Pass: In this step, the model starts with just one intercept term composed of the mean of the target (y) variables. Then the software divides the data into two distinct sets via a pair of basis functions. The area dividing point between these two areas is the knot referred to in equation 1. The position of this knot is calcuated such that it results in the maximum reduction in the RSS in equation 2. This process is repeated until the software has reached a maximum number of terms (as defined by the user prior to initializing the model), or until the change in RSS is too small to continue.

2. Backward Pass: The result of the first step will be a model that fits the data set extremely closely, however it is likely that the model will generalize well to other data. Thus, the backward pass (sometimes referred to as pruning) is applied. In this step, the model searches for the basis functions that contribute to the smallest increase in the goodness of fit. However, the RSS will always favor a model with more parameters, so a goodness of fit measure that penalizes additional parameters is needed. This is referred to as the Generalized Cross Validation GCV.

$$GCV = \frac{\sum_{i=1}^{n}(y_i - f(x_i))^2}{1 - \frac{C}{n}} \tag{5}$$

where $C = 1 + c * d$, n is the number of observations in the data, d is the number of independent basis functions and c is a penality for adding a basis function.

1.3 Illustrative Example of the Earth Software

To illustrate usage of the Earth software, a simple example data set was generated using the Data Painter tool from the Orange Machine Learning Python library. This data set is composed of an x variable which we attempt to use to predict the value of the y variable. We begin by loading the data, verifying the column names, and then inspect the data.

```
15
16  # Remember to add the "class" keyword to the third line, under the target variable
        . See here:
17  # http://orange.biolab.si/doc/tutorial/load-data/
18  data = orange.ExampleTable("painted_data_wo_outlier")
19  print data.domain.attributes
20  print data[:4]
```

Listing 1: Load and Inspect the Data

After reviewing the data, we convert it to *Numpy* arrays and plot the data using *Matplotlib*.

```
21 X, Y = data.to_numpy("A/C")
22
23 pl.plot(X, Y, ".r")
24 pl.title('Example Data Set')
25 pl.show()
```

Listing 2: Convert to Array and Plot

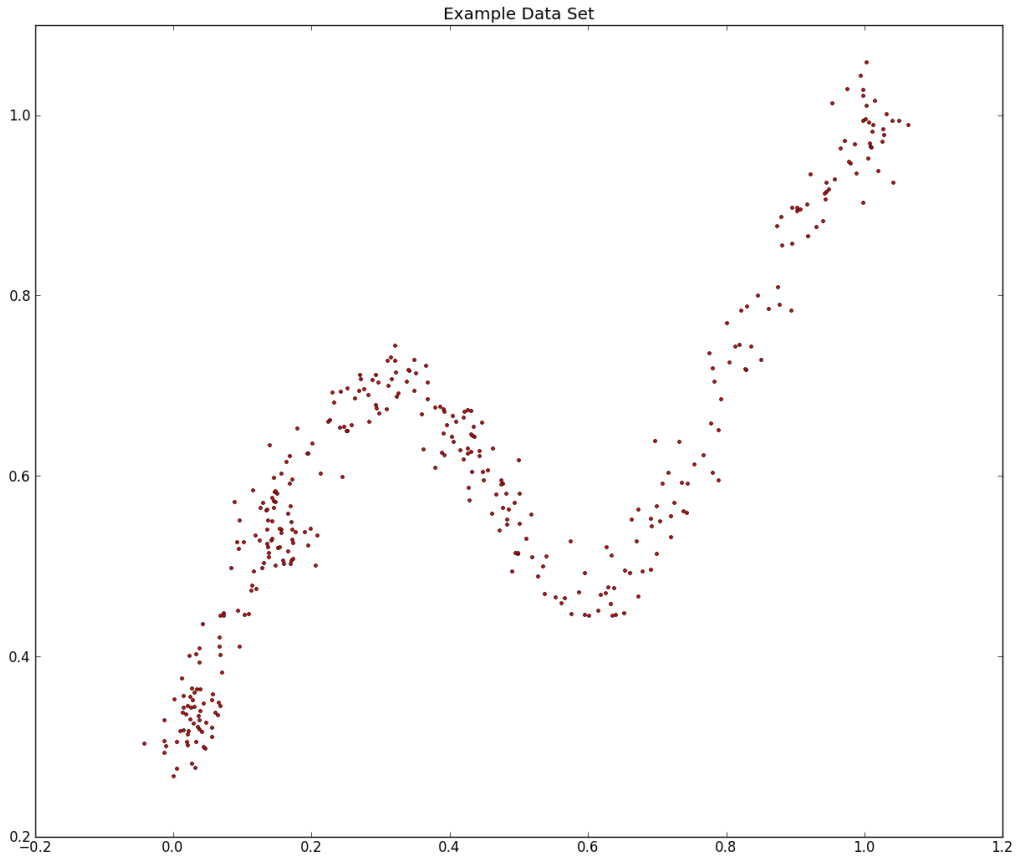

Fig. 1: Example Data

As can be seen from the graph, the data exhibits an irregular shape and the density of the points varies as one moves along the x-axis. Thus, this example should prove interesting to test to utility of the Earth software. Next, we use Earth to fit the data and examine the model.

```
27 earth_predictor = earth.EarthLearner(data)
28 print earth_predictor
```

Listing 3: Fit the Data

$$Y = 1.486 + 1.445 * \max(0, X - 0.744) - 1.590 * \max(0, 0.744 - X) - 1.818 * \max(0, X - 0.244) \quad (6)$$

$$+2.625 * \max(0, X - 0.640) - 0.682 * \max(0, X - 0.404) - 1.661 * \max(0, X - 0.979)$$

Next, we would like to see how well this model predicts the data. One strategy is to plot the values the Earth model predicts and compare them with the actual data. This is done by creating a *Numpy* array of x-values and then using the Earth model to predict the values of y.

```
earth_predictor = earth.EarthLearner(data)
print earth_predictor
linspace = numpy.linspace(min(X), max(X), 20)
predictions = [earth_predictor([s, "?"]) for s in linspace]
pl.plot(X, Y, ".r")
pl.plot(linspace, predictions, "-b")
pl.title('Example Data Set with Line Fit by MARS')
pl.show()
```

Listing 4: Compare the Predicted vs Actual Values and Plot the Results

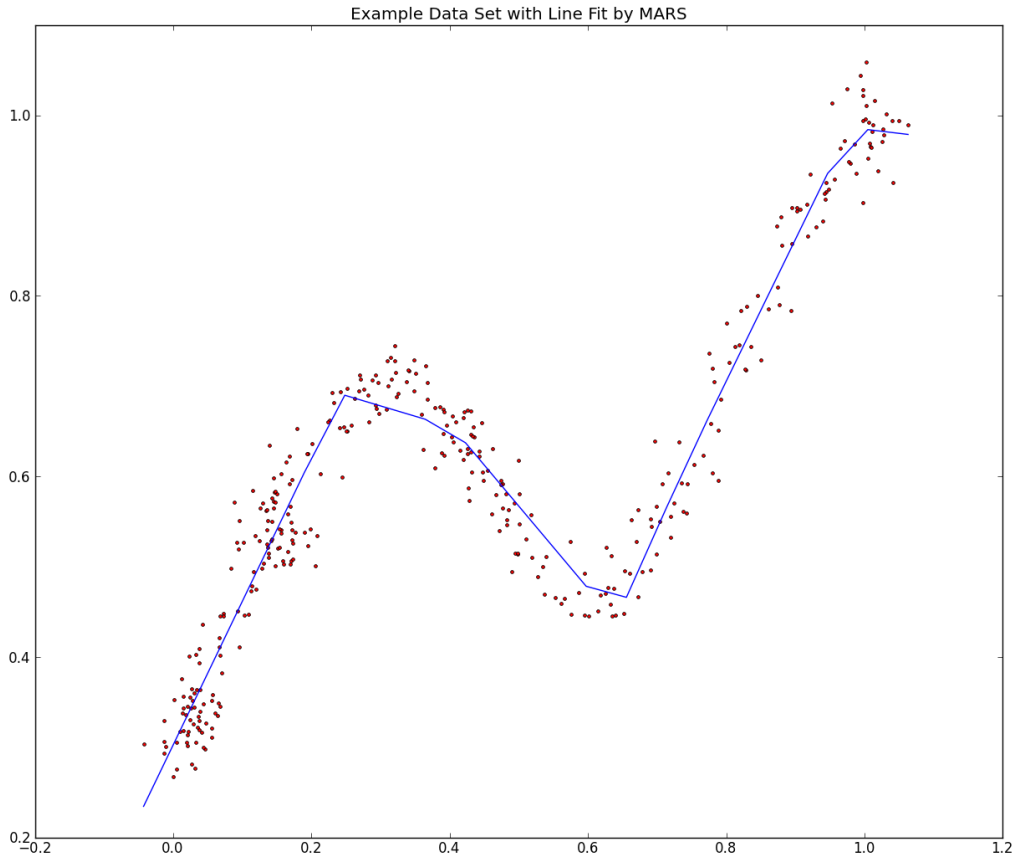

Fig. 2: Example Data Set with Line Fit by MARS