Polynomial Regression Model

Fabian Farelo, Autar Kaw, Jamie Trahan
University of South Florida
United States of America
kaw@eng.usf.edu

Introduction

Given n data points, (x_1, y_1), (x_2, y_2), ..., (x_n, y_n), use least squares method to regress the data to a m^{th} order polynomial.

$$y = a_0 + a_1 x + a_2 x^2 + ... + a_m x^m, \quad m < n \tag{1}$$

The residual at each data point is given by

$$E_i = y_i - a_0 - a_i x_i \; .... - a_m x_i^m \tag{2}$$

The sum of the square of the residuals is given by

$$Sr = \sum_{i=1}^{n} E_i^2 \tag{3}$$

To find the constants of the polynomial regression model, we put the derivatives with respect to a_i to zero, that is,

$$\partial \text{Sr} / \partial (a_0) = \sum_{i=1}^{n} [2 (y_i - a_0 - a_i x_i \; .... - a_m x_i^m)] (-1) = 0 \tag{4.a}$$

$$\partial \text{Sr} / \partial (a_1) = \sum_{i=1}^{n} [2 (y_i - a_0 - a_i x_i \; .... - a_m x_i^m)] (-x_i) = 0 \tag{4.b}$$

. . . .

. . . .

$$\partial \text{Sr} / \partial (a_m) = \sum_{i=1}^{n} [2 (y_i - a_0 - a_i x_i \; .... - a_m x_i^m)] (-x_i^m) = 0 \tag{4.m}$$

Setting those equations in matrix form gives

$$\begin{pmatrix} n & \sum x_i & ... & \sum x_i^m \\ \sum x_i & \sum x_i^2 & ... & \sum x_i^{m+1} \\ ... & ... & ... & ... \\ \sum x_i^m & \sum x_i^{m+1} & ... & \sum x_i^{2m} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ .. \\ a_m \end{pmatrix} = \begin{pmatrix} \sum y_i \\ \sum x_i y_i \\ ... \\ \sum x_i^m y_i \end{pmatrix}$$

The above simultaneous linear equations are solved for the $(m+1)$ constants, a_0, a_1, ... a_m. To learn more about polynomial regression see the Nonlinear Regression models worksheet.

Section 1: Input Data

Below are the input parameters to begin the simulation. This is the only section that requires user input. Once the values are entered, *Mathematica* will generate a polynomial regression model for the given data set.

- Number of Equations, *n*

```
n = 11
11
```

- Array of x values, "X"

```
X = {80, 40, 0, -40, -80, -120, -160, -200, -240, -280, -320}
{80, 40, 0, -40, -80, -120, -160, -200, -240, -280, -320}
```

- Array of y values, "Y"

```
Y = {6.47, 6.24, 6., 5.72, 5.43, 5.09, 4.72, 4.30, 3.83, 3.33, 2.76}
{6.47, 6.24, 6., 5.72, 5.43, 5.09, 4.72, 4.3, 3.83, 3.33, 2.76}
```

- Desired order of polynomial regression model

```
OrderPoly = 1
1
```

- Enter the lowest order of polynomial to check for optimum order:

```
LowOrder = 1
1
```

- Enter the highest order of polynomial to check for optimum order. **NOTE: *HighOrder* must be less than or equal to n-1.**

```
HighOrder = 6
6
```

Section 2: Polynomial Regression Model

In this section, the coefficient matrix "M" and right hand side vector "B" are calculated and subsequently used to determine the solution vector that contains the coefficients of the polynomial model $a_0, a_1, \ldots a_m$.

■ Determining the coefficient matrix, "M"

Creating the matrix size according to the order of the polynomial:

```
M = Array[0, {OrderPoly + 1, OrderPoly + 1}];
```

Determining each value of the first row of matrix "M":

```
M[[1, 1]] = n;
For[i = 2, i ≤ OrderPoly + 1, i++, M[[1, i]] = 0;
  For[j = 1, j ≤ n, j++,
   M[[1, i]] = N[M[[1, i]] + X[[j]]^(i - 1)]]];
```

Calculating the remaining values of the coefficient matrix:

```
For[i = 1, i ≤ OrderPoly + 1, i++,
 For[k = 2, k ≤ OrderPoly + 1, k++, M[[k, i]] = 0;
  For[j = 1, j ≤ n, j++,
   M[[k, i]] = N[M[[k, i]] + X[[j]]^(i + k - 2)]]]]
```

We now have the coefficient matrix, "M":

```
M // MatrixForm
```

$$\begin{pmatrix} 11 & -1320. \\ -1320. & 334400. \end{pmatrix}$$

■ Determining the right hand side vector, "B"

Creating the vector size according to the order of the polynomial model.

```
B = Array[0, {OrderPoly + 1}];
```

Finding the value of the array

```
B[[1]] = 0;

For[i = 1, i ≤ n, i++,
  B[[1]] = B[[1]] + Y[[i]]];
For[i = 2, i ≤ OrderPoly + 1, i++, B[[i]] = 0;
 For[j = 1, j ≤ n, j++,
  B[[i]] = N[B[[i]] + (X[[j]]^(i - 1)) * Y[[j]]]]]
```

Now, the right hand side vector "B"

```
B // MatrixForm
```

$$\begin{pmatrix} 53.89 \\ -4856.8 \end{pmatrix}$$

■ Calculating the coefficients of the model

Merging the matrices "M" and "B" to use *Mathematica* to reduce it:

```
F = Transpose[Append[Transpose[M], B]];
```

Selecting the last column with the coefficient's values.
This array contains the coefficients of the model as $a = [a_0, \ a_1, \ \ldots, \ a_m]$.

```
a = RowReduce[F];
a = a[[All, OrderPoly + 2]]
```

```
{5.99682, 0.00914773}
```

The polynomial regression model is as follows:

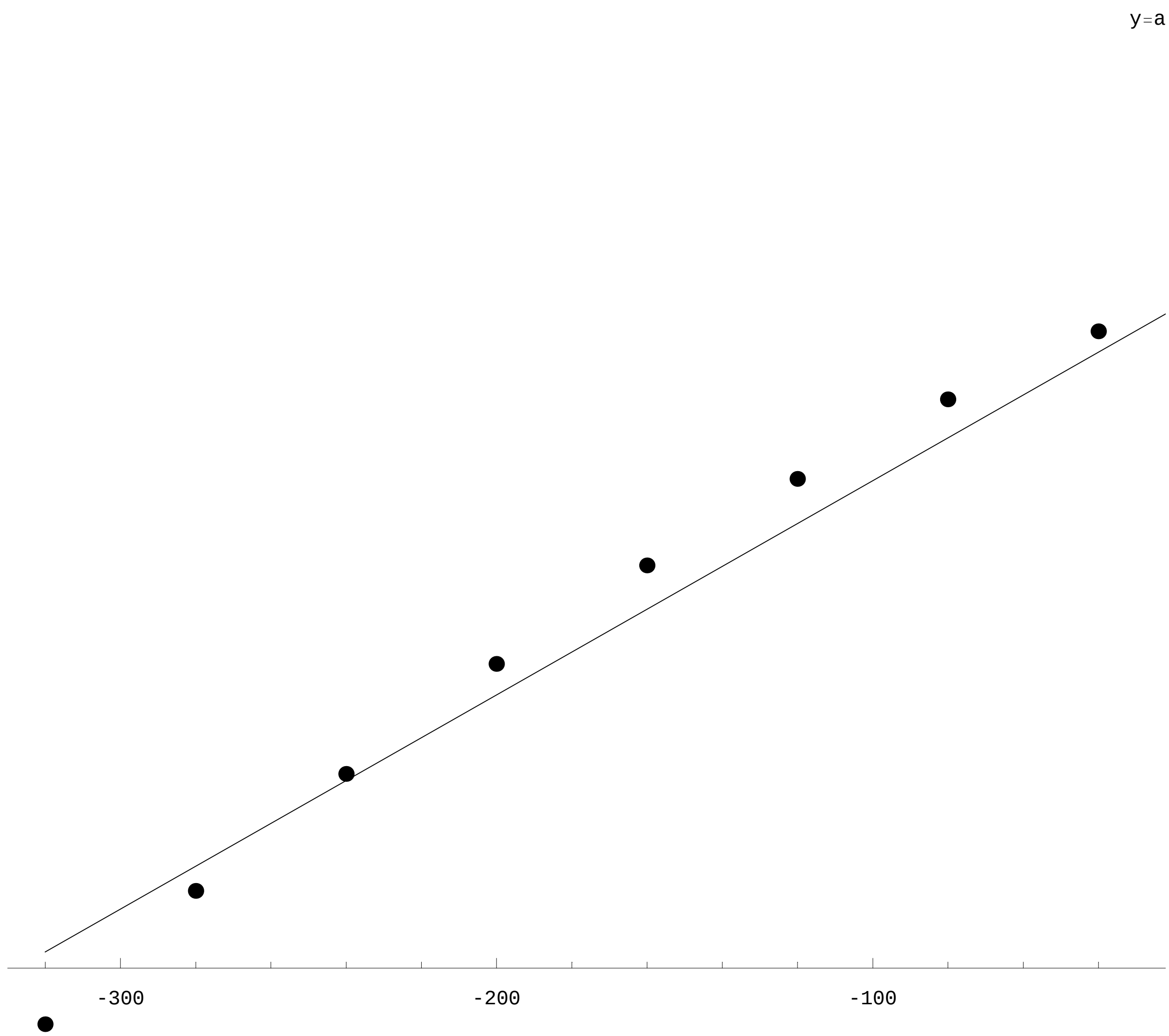
```
For[i = 1, i ≤ n, i++,
  y = a[[1]];
  For[j = 1, j ≤ OrderPoly, j++,
   y = y + a[[(j + 1)]] * x^j]];
Print["y= " y]
```

```
y=  (5.99682 + 0.00914773 x)
```

The plot shows the data points as well as the regression model.

```
Observed = Table[{X[[i]], Y[[i]]}, {i, 1, n}];
predicted = y;
ttl = Print["Polynomial Regression Model of order ", OrderPoly]
pt = ListPlot[Observed, PlotStyle → PointSize[0.01], DisplayFunction → Identity];
lin = Plot[y, {x, Min[X], Max[X]}, DisplayFunction → Identity];
Show[pt, lin, AxesLabel → {"X", "y=a0+a1x+a2x^2+...amx^m"},
 DisplayFunction → $DisplayFunction]
```

Polynomial Regression Model of order 1



- Graphics -

Section 3: Optimum Order

In this section we will determine the optimum order of the polynomial model by plotting the variance defined as

$$\frac{S_r}{n-(m+1)}$$

as a function of m, where n is the number of data points, S_r is the sum of the square of residuals and m is the order of the polynomial. The optimum order is considered as to be the one where the value of the variance $\frac{S_r}{n-(m+1)}$ is minimum or where its value is significantly decreasing.
In the following procedure, a polynomial regression model is calculated for each order specified in the *LowOrder* to *HighOrder* range. *Mathematica* then computes the variance of each model.The worksheet does not choose the order of the optimum polynomial for regression for you. Look at the plot of the variance as a function of the order of the polynomial. The optimum polynomial is one after which there is no statistical significant decrease in the variance.

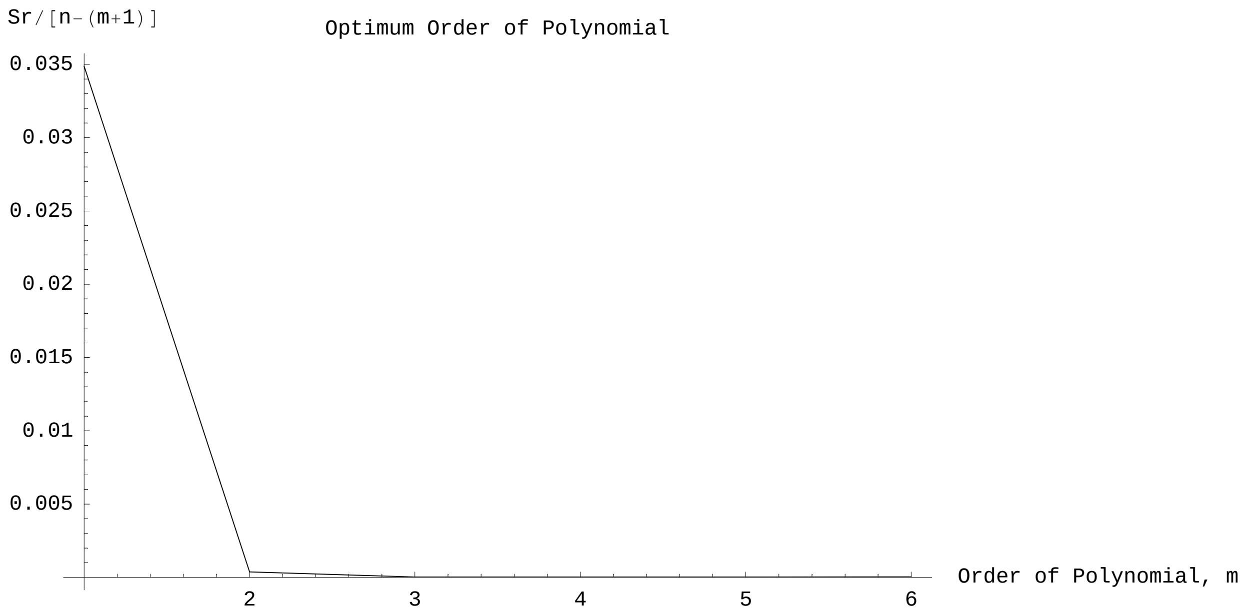
Many a times, the variance may show signs of decreasing and then increasing as a function of the order of the polynomial regression model. Such increases in the variance are normal as the variance is calculated as the ratio between the sum of the squares of the residuals and the difference between the number of data points and number of constants of the polynomial model. Both the numerator and denominator decrease as the order of the polynomial is increased. However, as the order of the polynomial increases, the coefficient matrix in the calculation of the constants of the model becomes more ill-conditioned. This ill-conditioning of the coefficient matrix results in fewer significant digits that can be trusted to be correct in the coefficients of the polynomial model, and hence artificially amplify the value of the variance.

```
Sr = Array[0, {HighOrder}];
(* In the procedure below, "m" is the order of polynomial being calculated,
 "M" is the coefficient matrix, "q" is the RHS vector, and "a" is the
  solution vector containing the coefficients of the polynomial model *)

For[m = LowOrder, m ≤ HighOrder, m++,
 (* Calculating the coefficient matrix "M" for the given order m *)
 M = Array[0, {m + 1, m + 1}];
 M[[1, 1]] = n;
 For[i = 2, i ≤ m + 1, i++, M[[1, i]] = 0;
  For[j = 1, j ≤ n, j++, M[[1, i]] = N[M[[1, i]] + X[[j]]^(i - 1)]]];
 For[i = 1, i ≤ m + 1, i++,
  For[k = 2, k ≤ m + 1, k++, M[[k, i]] = 0;
   For[j = 1, j ≤ n, j++,
    M[[k, i]] = N[M[[k, i]] + X[[j]]^(i + k - 2)]]]];
 (* Calculating the RHS vector for the given order, m *)
 q = Array[0, {m + 1}];
 q[[1]] = 0;
 For[i = 1, i ≤ n, i++,
  q[[1]] = q[[1]] + Y[[i]]];
 For[i = 2, i ≤ m + 1, i++, q[[i]] = 0;
  For[j = 1, j ≤ n, j++,
   q[[i]] = N[q[[i]] + (X[[j]]^(i - 1)) * Y[[j]]]]];
 (* Calculating the coefficients of the mth order polynomial model *)
 F = Transpose[Append[Transpose[M], q]];
 a = RowReduce[F];
 a = a[[All, m + 2]];
 (* Determining Sr *)
 Sr[[m]] = 0;
 For[i = 1, i ≤ n, i++, summ = 0;
  For[j = 1, j ≤ m, j++, summ = summ + a[[j + 1]] * X[[i]]^j];
  Sr[[m]] = Sr[[m]] + (Y[[i]] - (a[[1]] + summ))^2];]
(* Calculating the variance for the mth order Polynomial *)
var = Array[0, {HighOrder}];
For[i = LowOrder, i ≤ HighOrder, i++, var[[i]] = 0;
  var[[i]] = N[Sr[[i]] / (n - (i + 1))]];
Print["var =", var];
```

```
var ={0.0348722, 0.000380886, 0.0000273726, 0.0000261072, 0.0000308159, 0.0000324949}
```

```
ListPlot[var, PlotStyle → PointSize[0.02],
 PlotJoined → True, PlotLabel -> "Optimum Order of Polynomial",
 AxesLabel → {"Order of Polynomial, m", "Sr/[n-(m+1)]"}]
```



- Graphics -

Conclusion

Using *Mathematica*, we are able to regress a given data set to a polynomial model of the m^{th} order.

Question 1: Water is flowing through a pipe of radius 0.5 ft and flow velocity, v measurements are made from the center of the wall of the pipe as follows:

Radial Location, r (ft)	Velocity, v (ft/s)
0	10
0.08	9.7
0.17	8.9
0.25	7.5
0.32	5.6
0.42	3.1
0.50	0

a) Regress the data to

$$v = a_0\left(1 - \frac{r^2}{a^2}\right)$$

where a is the radius of the pipe.

b) Find the flow rate, $\dot{Q}$ through the pipe. (Hint: $\dot{Q} = \int 2\pi r v \, dr$, r=0..a)

Question 2: Thermal expansion coefficient of steel varies with temperature as given in the table below.

Temperature, T	Thermal expansion coefficient, $\alpha * E - 06$
80	6.47
60	6.36
40	6.24
20	6.12
0	6.00
−20	5.86
−40	5.72
−60	5.58
−80	5.43
−100	5.28
−120	5.09
−140	4.91
−160	4.72
−180	4.52
−200	4.30
−220	4.08
−240	3.83
−260	3.58
−280	3.33
−300	3.07
−320	2.76
−340	2.45

a) Regress the data to a second order polynomial.
$\alpha = a_0 + a_1 T + a_2 T^2$
b) Find the optimum order of polynomial for the regression model.
c) Find the reduction in the diameter of a steel cylinder of diameter 12.5" if it is cooled from a room temperature of 80°F to -180°F.

References

[1]Autar Kaw, *Holistic Numerical Methods Institute, http://numericalmethods.eng.usf.edu/nbm, See*
How does Nonlinear Regression work?