Get that Number

Approximate the inflection point from the interpolated equation.

Nathaniel Jones

KENNESAW STATE UNIVERSITY
COLLEGE OF COMPUTING AND SOFTWARE ENGINEERING
School of Data Science and Analytics

Faculty Advisors: Professor Michael Frankel & Professor Susan Hardy

$$Taylor\ Poly\ of degree\ n = ax^n + bx^{n-1} + \cdots + mx + n$$

Introduction

- Imputation can misrepresent the data if the tolerance of imputed values is set too high. If this tolerance is set too low, one risks losing information that may produce a better model.
- The point of inflection between the loss of information and the misrepresentation of the data is critical to ensure we have the maximum information we have to model and minimum error that model produces.
- This data was collected by running a SAS macro created by KSU professors to impute missing data. This researcher collected the input for the percent of observations that need imputing and the number of variables left in the data frame after the imputation macro completed.
- The code and data used to find the results of this project can be found by following the QR code in the top right corner and clicking on the GitHub icon.
- Overall, 43 observations were collected from this macro and used during this procedure and is included with the code file.

Methods

- **Least-Squares Approximation:** The least-squares approximation method determines an unknown relationship between two related variables, *x* and *y*, by minimizing error in approximation. In general, it uses a Taylor polynomial of degree *n* to approximate a cost function for the difference between the general polynomial and the observed *y* values. This will provide the expression used to derive the interpolating polynomial by computing the partial derivatives with respect to each weight. Set each derivative to zero before simplifying the expression by moving all values without an *x* to the right side, resulting in a matrix of *n* equations with *n* unknown values: the coefficients of the general Taylor polynomial. The leading entry in this special *nxn* matrix is the data's sum raised to the 2*n* power, and the following entries perform the same summation with exponent values one less than their adjacent entries to the left and above. This is the iterative process that the user can code into Python. One can then find the multiplicative product between the inverse of the calculated matrix and the residual matrix to calculate the coefficient weights of the interpolating polynomial.
- **Adjusted R-Square:** The adjusted R-squared can be used to determine if the addition of a parameter is useful to model. This value can be found by summing the squared error in prediction, weighting this value by the difference between the sample size and number of parameters, and dividing it by the mean squared error, weighted by the sample size minus one. This report found that the highest adjusted R-Squared (.9919) was achieved by the polynomial of degree 10 (Table 1).
- **Second Derivative:** A function's concavity, or curvature, is used to determine the direction of the function by the sign of the output at a given *x*. If the sign is negative, then the rate of change for the function is decreasing causing the curve to be concave down; if it is positive, then the curve will be concave up. An inflection point of the function is the point at which the sign changes from one to the other, or when the rate of change is equal to zero. This concept employs the data, the interpolated polynomial, and the bisection method to quickly find local points of inflection over a range of interest.
- **Bisection Method:** The bisection method helps determine the root, or zero, of a given function over a given range. This method utilizes the differing in the signs of the values on either side of zero to converge on the value of interest in the range of the function. This method employs the interpolated polynomial and its second derivative to quickly find local points of inflection in the given range. The midpoint is calculated alongside the evaluation of the midpoint and righthand endpoint by the second derivative of the function. We then move the endpoint that produces a value of similar signs to the value found by the midpoint. In this case, we used the range of values between 0.35 and 0.55 to determine a local point of inflection at 0.4382 (Figure 3) for the interpolated polynomial of degree 10. This allows the researcher to maximize the usefulness of a logistical model that will use the data imputed at a cut off point.

Figure 4: Matrix equations to calculate the weights of the interpolating polynomial of degree 2

$$\begin{bmatrix} a \\ b \\ c \\ \vdots \\ n \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^{n} (y_i x_i^n) \\ \vdots \\ \sum_{i=0}^{n} (y_i x_i) \\ \sum_{i=0}^{n} (y_i) \end{bmatrix} \begin{bmatrix} \sum_{i=0}^{n} (x_i^{2n}) & \sum_{i=0}^{n} (x_i^{2n-1}) & \cdots & \sum_{i=0}^{n} (x_i^{n}) \\ \sum_{i=0}^{n} (x_i^{2n-1}) & \sum_{i=0}^{n} (x_i^{2n-2}) & \cdots & \sum_{i=0}^{n} (x_i^{n-1}) \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=0}^{n} (x_i^{n}) & \sum_{i=0}^{n} (x_i^{n-1}) & \cdots & \sum_{i=0}^{n} (x_i^{0}) \end{bmatrix}^{-1}$$

Results & Recommendations

- Through least-squares approximation, a Python code interpolated the weights by solving the matrix equation in Figure 4. The weights for the first 50 polynomials were interpolated, and it was found that the highest adjusted R-squared value came from the 10th degree polynomial.
- To quickly find an inflection point, the bisection method uses the second derivative of the interpolated polynomial and the range of values between 0.35 and 0.55. After 12 iterations, a sufficiently small tolerance was achieved at the point 0.4382 (Figure 3).
- Extreme care must be taken when handling missing values to ensure misrepresentation and data loss are minimized. This method takes several precautions to ensure that the imputed data is representative of the original observations.

Table 1: Adjusted R-Squared for the interpolating polynomials.

Polynomial Degree	Adjusted R-Squared
1	0.8918
2	0.9042
3	0.9219
4	0.9201
5	0.9679
6	0.9896
7	0.9894
8	0.9895
9	0.9900
10	0.9919
11	0.9057

Figure 1: Scatterplot for the percent of data that needs to be imputed by the number of variables after imputation of the data.

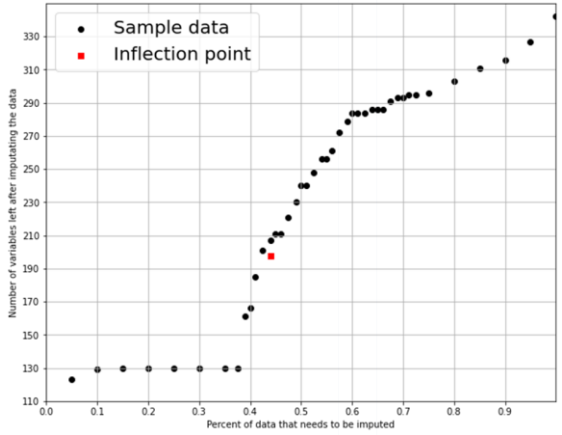


Figure 2: Scatterplot for the percent of data that needs to be imputed by the number of variables after imputation with the interpolating polynomial of degree 10.

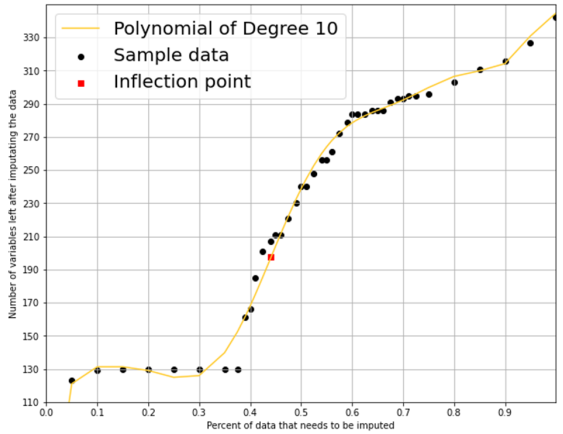
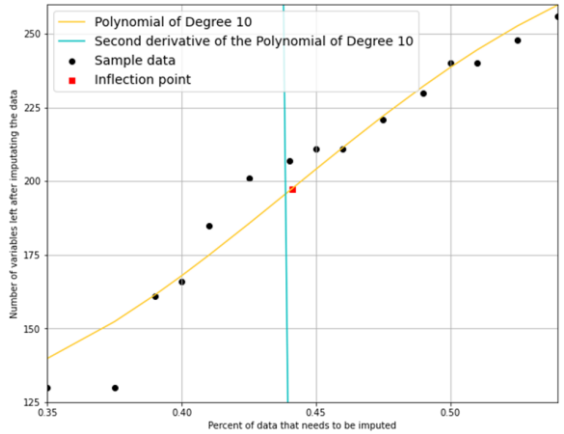


Figure 3: Scatterplot for the percent of data that needs to be imputed by the number of variables after imputation with the interpolating polynomial of degree 10 and its second derivative.



Least-Squares Approximations Computational Procedure

Data →

Map the Data:

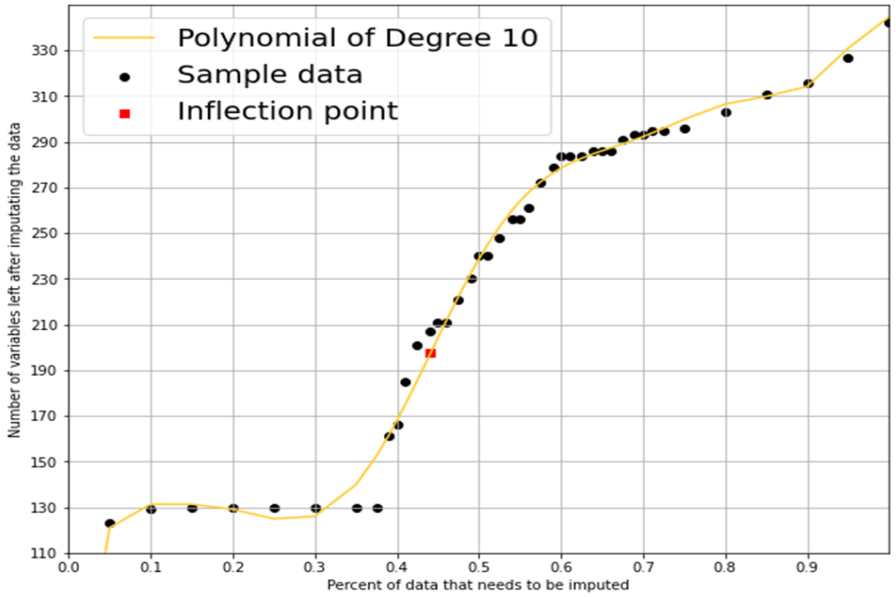
for i in range $(n+1)$:

  for i in range $(n+1)$:

    $A_{ij} = \Sigma(x^{2n-j-i})$

$$\boldsymbol{A_{ij} = \Sigma(x^{2n-j-i})}$$

$$\begin{pmatrix} \Sigma\left(x_{(0,0)}{}^{2n-0-0}\right) & \Sigma\left(x_{(1,0)}{}^{2n-1-0}\right) & \dots & \Sigma\left(x_{(n,0)}{}^{2n-n-0}\right) \\ \Sigma\left(x_{(0,1)}{}^{2n-0-1}\right) & \Sigma\left(x_{(1,1)}{}^{2n-1-1}\right) & \dots & \Sigma\left(x_{(n,1)}{}^{2n-n-1}\right) \\ \vdots & \vdots & & \vdots \\ \Sigma\left(x_{(0,n)}{}^{2n-0-n}\right) & \Sigma\left(x_{(1,n)}{}^{2n-1-n}\right) & \dots & \Sigma\left(x_{(n,n)}{}^{2n-n-n}\right) \end{pmatrix}$$



$$\begin{bmatrix} \sum_{i=0}^{n}(x_i^{2n}) & \sum_{i=0}^{n}(x_i^{2n-1}) & \dots & \sum_{i=0}^{n}(x_i^{n}) \\ \sum_{i=0}^{n}(x_i^{2n-1}) & \sum_{i=0}^{n}(x_i^{2n-2}) & \dots & \sum_{i=0}^{n}(x_i^{n-1}) \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=0}^{n}(x_i^{n}) & \sum_{i=0}^{n}(x_i^{n-1}) & \dots & \sum_{i=0}^{n}(x_i^{0}) \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=0}^{n}(y_i x_i^{n}) \\ \vdots \\ \sum_{i=0}^{n}(y_i x_i) \\ \sum_{i=0}^{n}(y_i) \end{bmatrix} = \begin{bmatrix} a \\ b \\ c \\ \vdots \\ n \end{bmatrix}$$

The Bisection Method road map

Initial Range of Interest from a to b

Success Check:

Step 1. $m = \frac{a+b}{2}$

Step 2. If $f'(m) - 0$ is sufficiently close to zero,

Then return (go right).

Else, yield (go left).

Inflection point found

Recursion:

1. If $f'(m) * f'(b) < 0$,
   Then the inflection point is between m and b.
   Else, the inflection point is between a and n.
2. Move the endpoint to the midpoint to zoom in on the interval.

Family of Partial Derivatives

$$Error = \Sigma(y - (ax^n + bx^{n-1} + \ldots + mx + n))^2$$

$$\left(\frac{dG}{da}\right) =$$

$$\left(\frac{dG}{db}\right) =$$

$$\vdots$$

$$\left(\frac{dG}{dm}\right) =$$

$$\left(\frac{dG}{dn}\right) =$$

$$\sum_{i=0}^{n}\left(a\left(x_i^{2n}\right) + b\left(x_i^{2n-1}\right) + \cdots + m\left(x_i^{n+1}\right) + n\left(x_i^{n}\right) - \left(y_i x_i^{n}\right)\right) = 0$$

$$\sum_{i=0}^{n}\left(a\left(x_i^{2n-1}\right) + b\left(x_i^{2n-2}\right) + \cdots + m\left(x_i^{n}\right) + n\left(x_i^{n-1}\right) - \left(y_i x_i^{n-1}\right)\right) = 0$$

$$\vdots \qquad\qquad \vdots$$

$$\sum_{i=0}^{n}\left(a\left(x_i^{n+1}\right) + b\left(x_i^{n}\right) + \cdots + m\left(x_i^{2}\right) + n\left(x_i\right) - \left(y_i x_i\right)\right) = 0$$

$$\sum_{i=0}^{n}\left(a\left(x_i^{n}\right) + b\left(x_i^{n-1}\right) + \cdots + m\left(x_i\right) + n\left(x_i^{0}\right) - \left(y_i\right)\right) = 0$$

Matrix in Figure 4

Plot of the first 11 polynomials found