Journal of Advanced Zoology
ISSN: 0253-7214
Volume 44 Issue S-8 Year 2023 Page 18:22

Exploring Population Dynamics in Nashik District: Applying Polynomial Extrapolation

S.S. Jadhav[1], R. O. Parmar[2], N. K. Patil[3], S.G. Pagare[4], M.A. Joshi[5], S.L. Khairnar[5*]

[1]*Department of Mathematics, Sundarrao More College of Arts, Commerce & Science, Poladpur*
[2]*Department of Geography, Changu Kana Thakur Arts, Commerce and Science College New Panvel*
[3]*JVM'S Mehta Degree College, Airoli, Navi Mumbai*
[4]*Department of Education, Brahma Valley Educational College of Anjaneri, Trimbakeshwar (Nashik)*
[5*]*Department of Mathematics, Changu Kana Thakur Arts, Commerce and Science College New Panvel*

***Corresponding Author:** S.L. Khairnar*
Email Address: sagarkhairnar1604@gmail.com

Abstract

In this study we have investigates the impact of polynomial degree selection on data fitting accuracy in analyzing population trends within Nashik District. Through a series of figures, it becomes evident that lower-degree polynomials, including linear and quadratic models, inadequately match the dataset's complexity. However, with escalating polynomial degrees, a notable improvement in fitting effectiveness emerges. This analysis highlights the critical role of selecting an appropriate polynomial degree in accurately representing underlying trends. While higher-degree polynomials offer improved fitting, the risk of overfitting, especially with smaller datasets, necessitates a delicate balance between complexity and accuracy. Understanding dataset characteristics is pivotal in determining the optimal polynomial degree for effective representation and prediction of population trends in Nashik District.
Keywords: Polynomial Degree, Data Fitting Accuracy, Population Trends, Nashik District, Overfitting

Introduction:

In numerical analysis, spline interpolation utilizes a particular type of piecewise polynomial, known as a spline, as the interpolant. Rather than employing a single high-degree polynomial to fit all values simultaneously, spline interpolation employs low-degree polynomials to interpolate small subsets of the values. This method is favoured over polynomial interpolation due to its ability to minimize interpolation errors, even with the use of low-degree polynomials for the spline. Additionally, spline interpolation circumvents Runge's phenomenon, a challenge associated with high-degree polynomials that leads to oscillations between points during interpolation.

Literature Review:

Harju employed an FIR polynomial predictor for data with missing samples, comparing its performance with a computationally lighter algorithm based on decision-driven recursion in predicting a sinusoidal signal corrupted by impulsive and Gaussian noise(Harju, 1997). Ghazali et al. demonstrated the use of a ridge polynomial neural network for forecasting financial time series trends(Ghazali et al., 2006). Hussain et al. proposed a novel higher-order pipelined neural network known as the polynomial pipelined neural network(Hussain et al., 2008). Zaw and Naing forecasted rainfall in Myanmar using MPR and MLR models, concluding that the MPR model delivered more accurate results(Zaw & Naing, 2009). Bahadori applied spline interpolation for extracting natural gas from water-saturated underground reservoirs(Bahadori, 2011). Additionally, Bahadori & Nouri developed a straightforward model to estimate the critical oil rate for bottom water coning in anisotropic and homogeneous formations with the well completed from the top of the formation(Bahadori & Nouri, 2012). Physicists often explore complex, nonlinear models with numerous unknown or adjustable parameters to interpret experimental data. Ostertagová utilized a polynomial regression model to establish the relationship between strains and drilling, analyzing the data using a computer program(Ostertagová, 2012). Zjavka developed an accurate short-term wind speed forecasting model crucial for renewable energy power generation planning, particularly in grid systems(Zjavka, 2015). In 625 AD, Indian astronomer and mathematician Brahmagupta introduced a method for second-order interpolation of the sine function and later devised a technique for interpolating unequal-interval data(Lavanya & Achireddy, 2016). Wu et al. proposed CryptoNets, a scheme for prediction on encrypted data using neural networks, evaluating the expressiveness of PPoly activations and discussing the accuracy-efficiency tradeoff (Wu et al., 2018).

Methodology:

A polynomial with degree n means a linear combination of the term $x^i, i = 0,1,2,\ldots,n$. For every set of data point there is a unique interpolation polynomial with degree= no. of points -1. From the complicated mathematical model or a function f(x) we can get the data points. If we get an interpolation polynomial the we can easily replace the original model that is the function by the interpolation polynomial for the purpose of analysis and design.
We can write the n points $\{(x_1, p(x_1)), (x_2, p(x_2)), \ldots, (x_n, p(x_n))\}$ in the form of polynomial of degree (n-1).

$$p(x) = \beta_{n-1}x^{n-1} + \cdots + \beta_2 x^2 + \beta_1 x + \beta_0$$

Further this polynomial can be expressed as $AX = B$. A is the $n \times n$ matrix which is known as Vandermonde matrix, X and B is $n \times 1$ matrix given as

$$A = \begin{bmatrix} x_1^{n-1} & \cdots & x_1 & 1 \\ x_2^{n-1} & \ldots & x_2 & 1 \\ \vdots & \ddots & \vdots & \vdots \\ x_n^{n-1} & \cdots & x_n & 1 \end{bmatrix}, X = \begin{bmatrix} \beta_{n-1} \\ \beta_{n-2} \\ \vdots \\ \beta_1 \\ \beta_0 \end{bmatrix}, B = \begin{bmatrix} p(x_1) \\ p(x_2) \\ \vdots \\ p(x_n) \end{bmatrix}$$

The solution for accurate interpolation in systems with roundoff errors involves understanding the trade-offs between high-order interpolating polynomials and their potential pitfalls.
When dealing with large datasets, using high-order interpolating polynomials might lead to overfitting. Using a single high-degree polynomial to interpolate between points can pass exactly through these points but may not generalize well between them, especially if the number of points is limited.
To ease this, employing multiple low-order polynomials for piecewise interpolation—known as spline interpolation—can yield better results. Each polynomial within this approach is valid within an interval between two or more points and maintains the same degree but with different coefficients.
In spline interpolation, the points where adjacent splines meet is called knots. At these knots, the derivative may not be continuous, leading to abrupt changes in slope. To ensure a smoother interpolation function, higher-order polynomials can be used to enforce continuity of the second derivative at each knot point. This helps to improve the smoothness of the underlying function being interpolated.
The key elements of the spline method revolve around determining the lowest-order polynomial spline s_i that passes through two adjacent interpolating points $(x_i, f(x_i))$ and $(x_{i+1}, f(x_{i+1}))$. This spline should satisfy specific conditions:
Interpolation: The spline s_i must interpolate the given points $(x_i, f(x_i))$ and $(x_{i+1}, f(x_{i+1}))$.
Continuity of First Derivative: The slope (first derivative) of s_i at x_i should be equal to the slope of the previous spline s_{i-1} at x_i

Continuity of Second Derivative: The concavity (second derivative) of s_i at x_i must be the same as that of the previous spline s_{i-1} at x_i.
These conditions result in four constraints that need to be satisfied to determine the characteristics of the spline s_i accurately. These constraints ensure a smooth and continuous transition between adjacent splines while maintaining the desired properties at the interpolation points and providing a coherent overall interpolation function.

$$s_i(x_i) = f(x_i)$$

$$s_i(x_{i+1}) = f(x_{i+1})$$

$$s'_i(x_i) = s'_{i-1}(x_i)$$

$$s_i''(x_i) = s_{i-1}''(x_i)$$

Where $'$ denotes the first derivative w.r.t. x and $''$ denotes the second derivative w.r.t. x.

The polynomial that fulfils these four constraints necessitates a minimum of four degrees of freedom, typically corresponding to a 3rd-order polynomial, commonly referred to as a cubic polynomial. Therefore, s_i has the form,

$$s_i(x) = \beta_{i,3}x^3 + \beta_{i,2}x^2 + \beta_{i,1}x + \beta_{i,0} \quad \forall i = 0,1,2, \dots n-1$$

We need $n-1$ splines to accommodate and interpolate n data points.

In order to calculate all $4(n-1)$ spline coefficients, solving $4(n-1)$ equations is necessary. Specifically, when dealing with cubic splines involving three data points (x_1, x_2, x_3), we determine that 2 cubic splines are required for these three points.
After solving we get the following system.

$$\begin{bmatrix} x_1^3 & x_1^2 & x_1 & 1 & 0 & 0 & 0 & 0 \\ x_2^3 & x_2^2 & x_2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x_2^3 & x_2^2 & x_2 & 1 \\ 0 & 0 & 0 & 0 & x_3^3 & x_3^2 & x_3 & 1 \\ 3x_2^2 & 2x_2 & 1 & 0 & -3x_2^2 & -2x_2 & -1 & 0 \\ 3x_2 & 1 & 0 & 0 & -3x_2 & -1 & 0 & 0 \\ 3x_1^2 & 2x_1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3x_3^2 & 2x_3 & 1 & 0 \end{bmatrix} \begin{bmatrix} \beta_{1,3} \\ \beta_{1,2} \\ \beta_{1,1} \\ \beta_{1,0} \\ \beta_{2,3} \\ \beta_{2,2} \\ \beta_{2,1} \\ \beta_{2,0} \end{bmatrix} = \begin{bmatrix} p(x_1) \\ p(x_2) \\ p(x_2) \\ p(x_3) \\ 0 \\ 0 \\ p'(x_1) \\ p'(x_3) \end{bmatrix}$$

For the spline polynomial interpolation, we have used MATLAB. We have collected the data from the government website (*Census Tables | Government of India*, n.d.).

Results and Conclusion:

From Figure 1, its clear that linear polynomials don't perfectly match the data. Similarly, Figure 2 illustrates that degree 2 polynomials also fall short of perfect alignment with the data. As the polynomial degree increases (Figure 3 to Figure 8), the data fit into the model more effectively. In Figure 10, it is noticeable that the degree 1 polynomial is inadequate for fitting, while there is a slight disparity between the fittings of degree 2 and 3 polynomials. Figure 10 further emphasizes that as the degree of the polynomial increases, the accuracy of the fitting also improves. As compared to the degree 1,2,3 and 4 polynomial the degree 4 polynomial i.e. quartic polynomial gave the best approximation for the data. Linear and quadratic polynomial not gave the perfect approximation of the data. By using quartic polynomial, we can get more reliable predictions as compared to linear, quadratic polynomial. This example shows that extrapolating data using polynomials of even modest degree is dicey and defective.

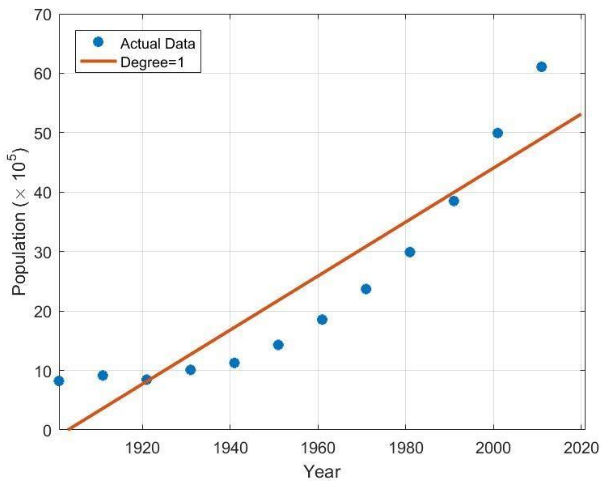

Figure 1

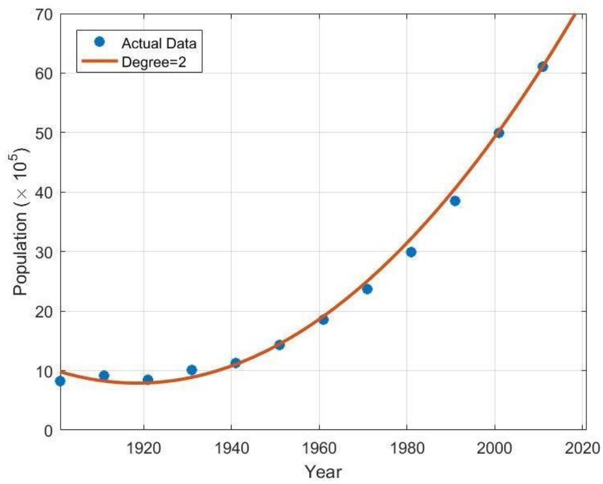

Figure 2

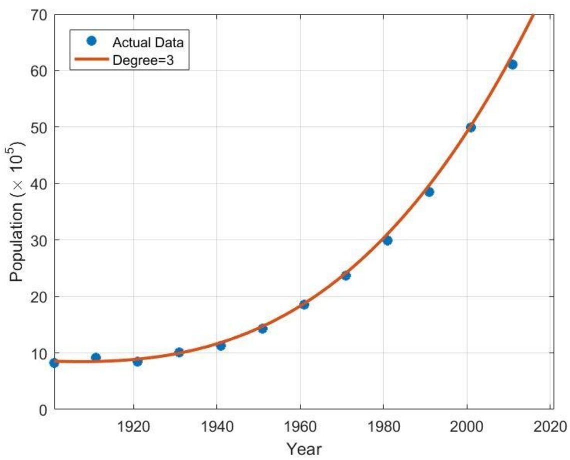

Figure 3

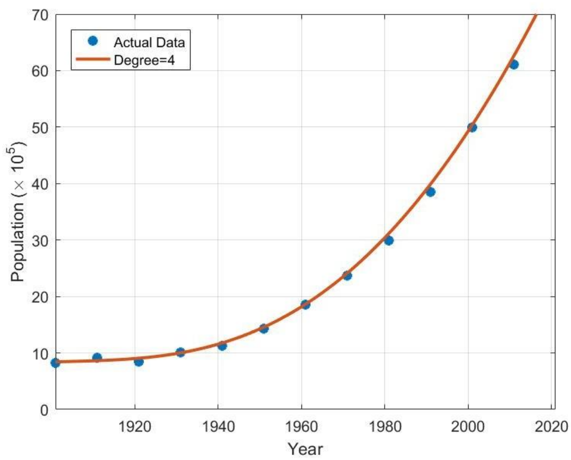

Figure 4

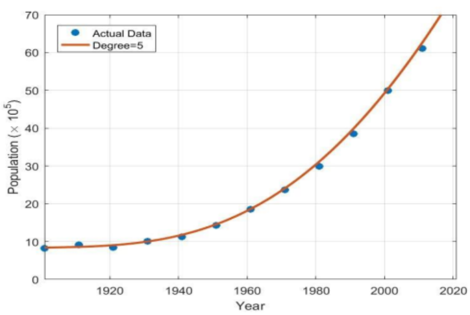

Figure 5

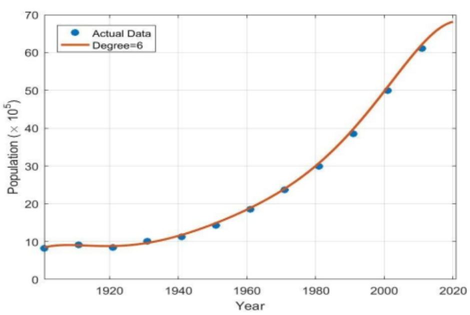

Figure 6

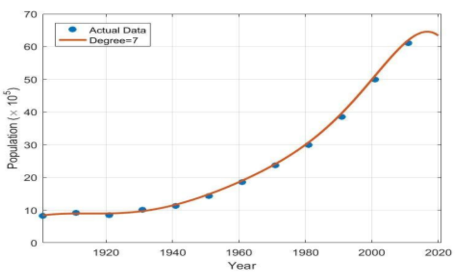

Figure 7

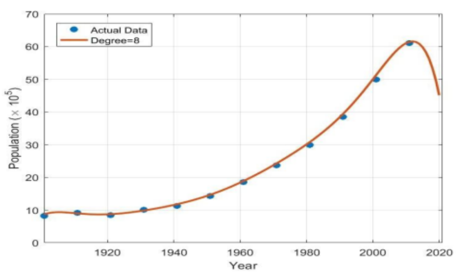

Figure 8

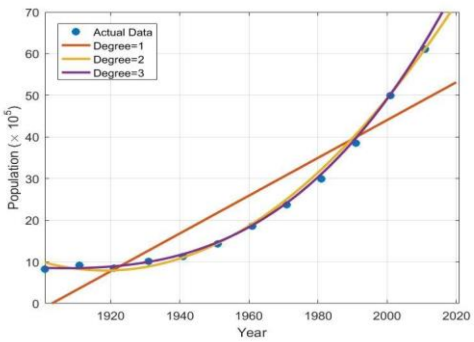

Figure 9

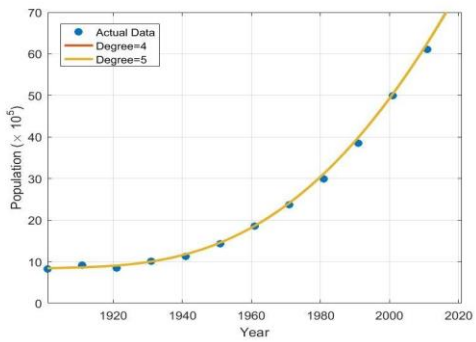

Figure 10

Conclusion

In conclusion, the analysis of polynomial fitting reveals that the choice of polynomial degree significantly impacts the accuracy of fitting data. Lower-degree polynomials, such as linear or quadratic, may not capture the intricacies of the data adequately, resulting in imperfect fits. As the polynomial degree increases, the models tend to better approximate the data, reflecting improved accuracy in fitting. However, higher-degree polynomials, while providing better fitting, may also risk overfitting the data, particularly with smaller datasets, leading to reduced generalization ability. Therefore, selecting an appropriate polynomial degree requires a careful balance between capturing the nuances of the data and avoiding excessive complexity. It is essential to assess the trade-offs between model complexity and accuracy to determine the most suitable polynomial degree for effectively representing and predicting the underlying patterns within the dataset. Additionally, considering the context, size, and nature of the dataset is crucial in making informed decisions about the polynomial degree for optimal fitting and predictive performance.

Reference:

1. Bahadori, A. (2011). Prediction of moisture content of natural gases using simple Arrhenius-type function. *Central European Journal of Engineering*, *1*(1), 81–88.
2. Bahadori, A., & Nouri, A. (2012). Prediction of critical oil rate for bottom water coning in anisotropic and homogeneous formations. *Journal of Petroleum Science and Engineering*, *82*, 125–129.
3. *Census tables | Government of India*. (n.d.). Retrieved January 6, 2024, from https://censusindia.gov.in/census.website/data/census-tables
4. Ghazali, R., Hussain, A., & El-Deredy, W. (2006). Application of Ridge Polynomial Neural Networks to Financial Time Series Prediction. *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, 913–920. https://doi.org/10.1109/IJCNN.2006.246783
5. Harju, P. T. (1997). Polynomial prediction using incomplete data. *IEEE Transactions on Signal Processing*, *45*(3), 768–770. https://doi.org/10.1109/78.558500
6. Hussain, A. J., Knowles, A., Lisboa, P. J. G., & El-Deredy, W. (2008). Financial time series prediction using polynomial pipelined neural networks. *Expert Systems with Applications*, *35*(3), 1186–1199. https://doi.org/10.1016/j.eswa.2007.08.038
7. Lavanya, Y., & Achireddy, C. (2016). *APPLICATIONS OF INTERPOLATION*. *3*(5), 5.
8. Ostertagová, E. (2012). Modelling using Polynomial Regression. *Procedia Engineering*, *48*, 500–506. https://doi.org/10.1016/j.proeng.2012.09.545
9. Wu, W., Liu, J., Wang, H., Tang, F., & Xian, M. (2018). PPolyNets: Achieving High Prediction Accuracy and Efficiency With Parametric Polynomial Activations. *IEEE Access*, *6*, 72814–72823. https://doi.org/10.1109/ACCESS.2018.2882407
10. Zaw, W. T., & Naing, T. T. (2009). Modeling of Rainfall Prediction over Myanmar Using Polynomial Regression. *2009 International Conference on Computer Engineering and Technology*, *1*, 316–320. https://doi.org/10.1109/ICCET.2009.157
11. Zjavka, L. (2015). Wind speed forecast correction models using polynomial neural networks. *Renewable Energy*, *83*, 998–1006. https://doi.org/10.1016/j.renene.2015.04.054