Polynomial Regression Models

Sila Aydin, Elif Ilgin Yildiz, Huseyin Mert Ezer

POLYNOMIAL REGRESSION MODEL

- **Linearity in Parameters:** The model is linear with respect to the **coefficients** (beta), not the inputs (x).
- **Global Sensitivity:** Uses **global basis functions**, meaning a single data point change shifts the entire curve, not just a local section.
- **Feature Projection:** Solves non-linearity by mapping inputs to a **higher-dimensional space** where the data fits a linear hyperplane.
- **Vandermonde Instability:** The specific matrix structure causes **ill-conditioning** at high degrees (d > 10), leading to numerical failure.

Hypothesis function:

$$\hat{y}_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \cdots + \beta_d x_i^d$$

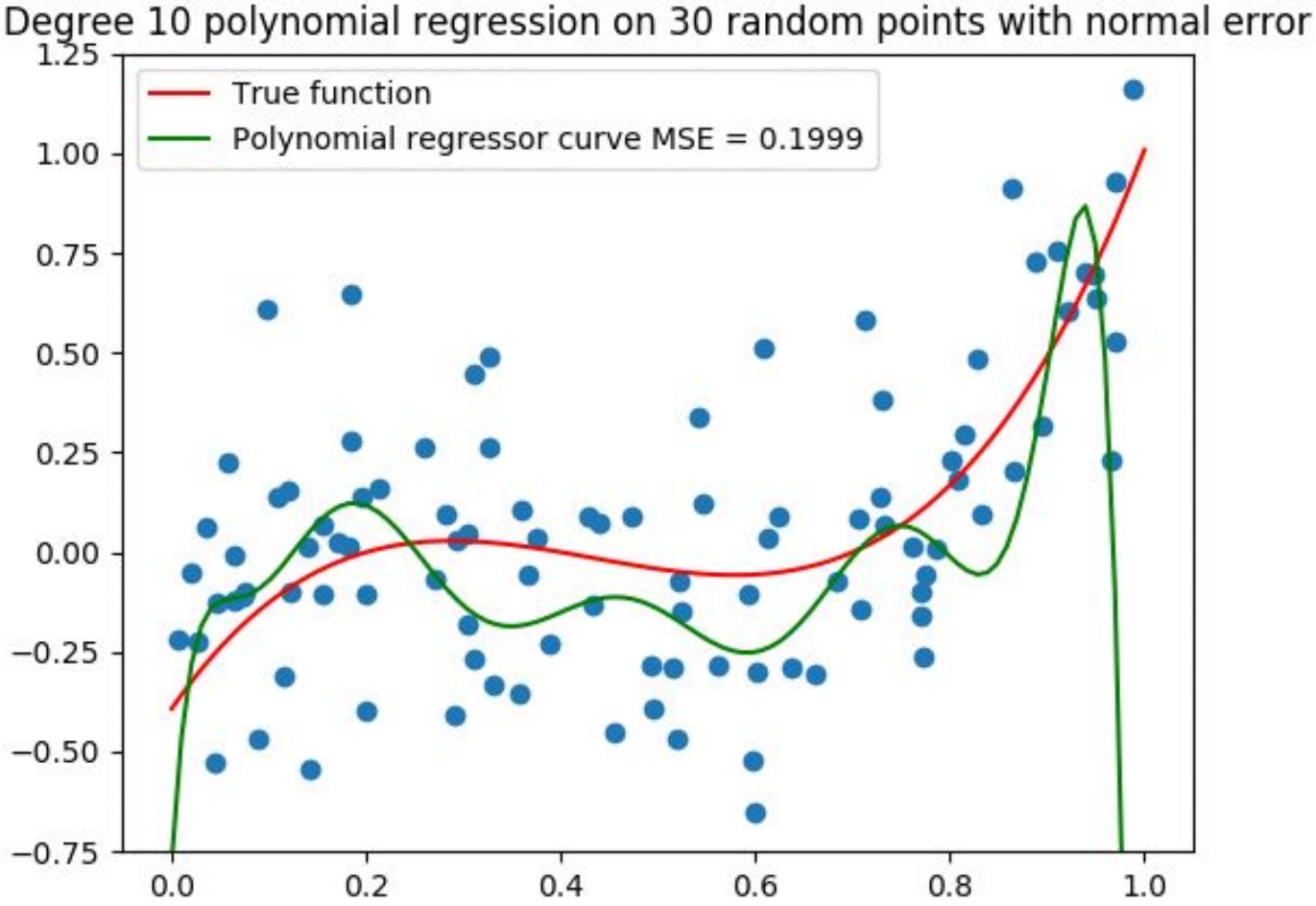
Matrix representation:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

$$\mathbf{X} = \begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^d \\ 1 & x_2 & x_2^2 & \dots & x_2^d \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_N & x_N^2 & \dots & x_N^d \end{bmatrix}$$

It is generally solved using the OLS –>

$$\min_{\boldsymbol{\beta}} ||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2^2$$


Degree 10 polynomial regression on 30 random points with normal error
True function
Polynomial regressor curve MSE = 0.1999
1.25
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
0.0
0.2
0.4
0.6
0.8
1.0

Pros and Cons of Polynomial Regression Model

Category	Pros	Cons
Speed	**Very Fast:** Once built, it gives answers instantly, which is perfect for running many tests quickly.	**Risk of Overfitting:** The model can try too hard to fit every training points, leading to wild errors.
Understanding	**Easy to Explain:** You can easily see which input matters most because it has a large/low coefficient (β).	**Limited Shapes:** It can only fit smooth, simple curve shapes (like straight lines or gentle parabolas). It **cannot** handle sharp corners, sudden drops, or complex wiggles in your results.
Coverage	**Smooth & Global:** It gives a continuous, smooth surface that covers the whole design area nicely.	**No Error Check:** Unlike Gaussian Model, it doesn't tell you how confident it is in its prediction (no built-in measure of uncertainty).

Pros and Cons of Using Polynomial Regression Model for Designs

Design	Pros	Cons
Full Factorial Design	**Perfect Orthogonality**: Variables are independent, leading to precise coefficient estimates.	**Exponential Cost**: Exponential growth, Highly inefficient in high dimensions.
Latin Hypercube Sampling	**Efficient & Scalable:** Covers the design space well with few runs. **Ideal for high-dimensional problems.**	**Not Perfectly Orthogonal**: Unlike FFD, the design matrix is not perfectly orthogonal, meaning estimates of main effects and interactions can sometimes be correlated.

R Code

```
train_data <- data.frame(
  y  = y_train, x1 = x1_train, x2 = x2_train, x3 = x3_train, x4 = x4_train,x5 = x5_train,
  x6 = x6_train)

surrogate_model <- lm(y ~ (x1 + x2 + x3 + x4 + x5 + x6)^2 + I(x1^2) + I(x2^2) + I(x3^2) +
    I(x4^2) + I(x5^2) + I(x6^2),
  data = train_data)

summary(surrogate_model)

test_data <- data.frame(x1 = x1_test,x2 = x2_test, x3 = x3_test, x4 = x4_test,x5 = x5_test,
  x6 = x6_test)

y_test_pred <- predict(surrogate_model, newdata = test_data)

rmse <- sqrt(mean((y_test_pred - y_test_true)^2))
```