Regression: Multi-Linear, Polynomial and Splines

ML2: AI Concepts and Algorithms (SS2025)

Faculty of Computer Science and Applied Mathematics
University of Applied Sciences Technikum Wien

FH University of Applied Sciences TECHNIKUM WIEN

Lecturer: Rosana de Oliveira Gomes
Author: S. Rezagholi, R.O. Gomes

Recap: Multi-Linear Regression

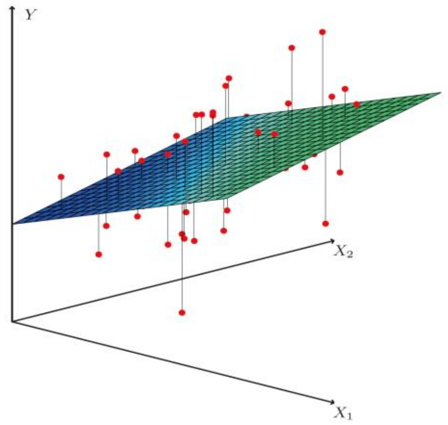
- A quantitative target variable **y** with **p different predictors** $x_1, x_2, ..., x_J$ is written in the form

$$Y = f(X_1, \ldots, X_J) + \epsilon = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_J X_J + \epsilon$$

$$Y = y_1, \ldots, y_n, \quad X = X_1, \ldots, X_J \text{ and } X_j = x_{1j}, \ldots, x_{ij}, \ldots, x_{nj}$$

ε = stochastic error term

- The parameters $\beta_0, \beta_1, ..., \beta_p$ are estimated using the least squares approach as in simple linear regression.



Example: sales based on multiple marketing sources

n observations

TV	radio	newspaper	sales
230.1	37.8	69.2	22.1
44.5	39.3	45.1	10.4
17.2	45.9	69.3	9.3
151.5	41.3	58.5	18.5
180.8	10.8	58.4	12.9

p predictors

Matrix Notation

Multi-Linear Regression in al algebraic form:

$$\mathbf{Y} = \mathbf{X}\beta + \epsilon$$

$$\mathbf{Y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \qquad \mathbf{X} = \begin{pmatrix} 1 & x_{1,1} & \dots & x_{1,J} \\ 1 & x_{2,1} & \dots & x_{2,J} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n,1} & \dots & x_{n,J} \end{pmatrix} \qquad \beta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_J \end{pmatrix}$$

FH University of Applied Sciences TECHNIKUM WIEN

Ordinary Least Squares Method (OLS):

Multi-Linear Regression: residual sum of squares in matrix and sum notation

$$RSS(\beta) = (\mathbf{Y} - \mathbf{X}\beta)^{\top}(\mathbf{Y} - \mathbf{X}\beta) = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

$$\mathbf{r} = \mathbf{Y} - \mathbf{X}\beta = \begin{bmatrix} y_1 - \hat{y}_1 \\ y_2 - \hat{y}_2 \\ \vdots \\ y_n - \hat{y}_n \end{bmatrix} \qquad r_i = y_i - \hat{y}_i$$

The **summation notation** provides a clear algebraic interpretation of the matrix expression.

OLS: minimize RSS in relation to beta

$$\frac{\partial RSS(\beta)}{\partial \beta} = 0$$

$$\boxed{\hat{\beta} = (\mathbf{X}^{\top}\mathbf{X})^{-1}\mathbf{X}^{\top}\mathbf{Y}}$$

FH University of Applied Sciences TECHNIKUM WIEN

Polynomial Regression

Motivation: how to represent nonlinear relationships?

Stone-Weierstrass theorem:
Any continuous nonlinear regression model can be realized as a polynomial regression model!

Model form:

Y target described in term of non-linear dependence of one variable X

Polynomials of degree d:

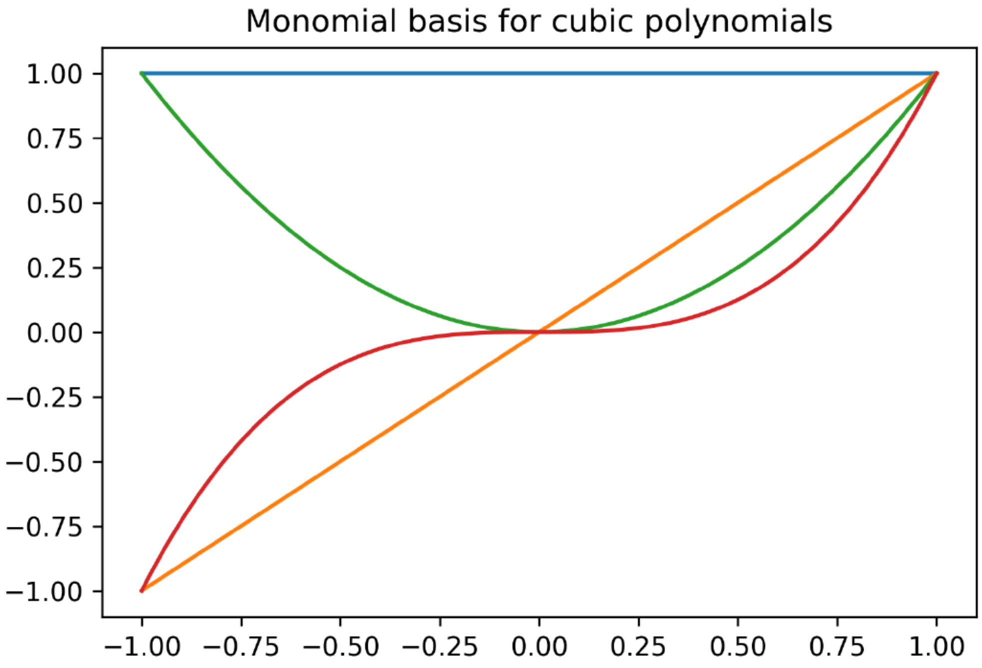
$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \cdots + \beta_d X^d + \epsilon$$

Monomials

Every polynomial of degree d is a linear combination of the following basis functions

$$\{x^0, x^1, x^2, \ldots, x^d\}$$

Blue: 0-th order (constant)
Orange: 1st order (linear)
Green: 2nd order (quadratic)
Red: 3rd order (cubic)



Polynomial Regression: Design Matrix

$$\mathbf{Y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}, \qquad \mathbf{X} = \begin{pmatrix} 1 & x_1^1 & \dots & x_1^M \\ 1 & x_2^1 & \dots & x_2^M \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & \dots & x_n^M \end{pmatrix}, \qquad \boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_M \end{pmatrix}.$$

Each polynomial is treated as a separate predictor

FH University of Applied Sciences TECHNIKUM WIEN

OLS Solution remains the same

$$\boxed{\hat{\beta} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{Y}}$$

Polynomial Regression: Generalizations

Just as in the case of linear regression with cross terms, polynomial regression is a special case of linear regression

polynomial models with multiple predictors {X_1, ..., X_J}

$$
\begin{aligned}
y = &\beta_0 + \beta_1 x_1 + \ldots + \beta_M x_1^M \\
&+ \beta_{M+1} x_2 + \ldots + \beta_{2M} x_2^M \\
&+ \ldots \\
&+ \beta_{M(J-1)+1} x_J + \ldots + \beta_{MJ} x_J^M
\end{aligned}
$$

polynomial models with multiple predictors {X_1, X_2} and cross terms

$$
\begin{aligned}
y = &\beta_0 + \beta_1 x_1 + \ldots + \beta_M x_1^M \\
&+ \beta_{1+M} x_2 + \ldots + \beta_{2M} x_2^M \\
&+ \beta_{1+2M} (x_1 x_2) + \ldots + \beta_{3M} (x_1 x_2)^M
\end{aligned}
$$

FH University of Applied Sciences TECHNIKUM WIEN

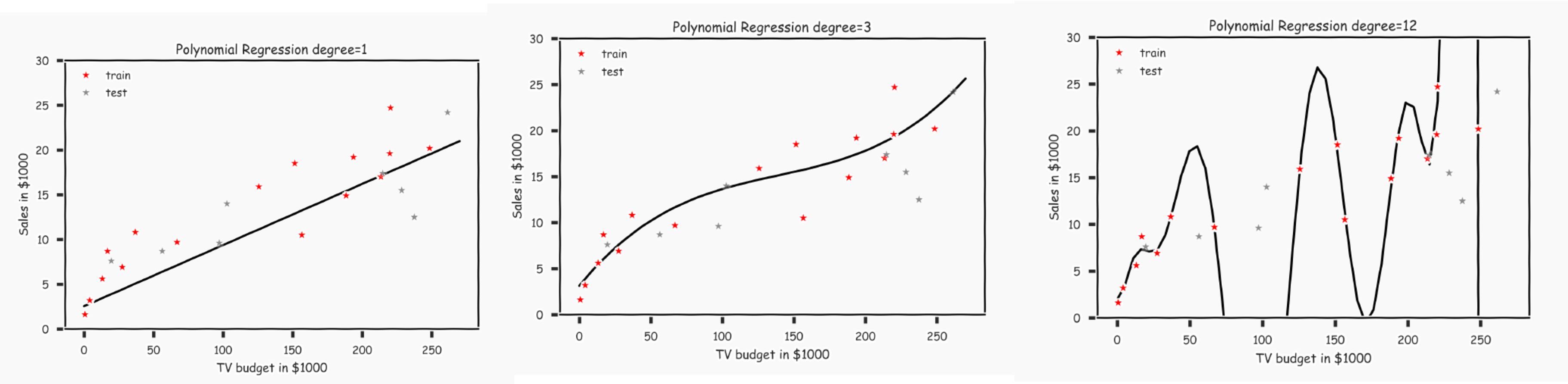
Each polynomial is treated as a separate predictor x_j^m

OLS method still holds

Overfitting

Unnecessarily complex model that captures the random noise in the observation (training data) and performs poor predictions in new data

Polynomials tend to oscillate too much



Polynomials are prone to overfitting

Overfitting

Common Scenarios:

A. Too many predictors:

- the feature space has high dimensionality
- the polynomial degree is too high
- too many cross terms are considered

B. The coefficients values are too **extreme**

Symptoms:

high R-squared (low error) in training data and poor performance on testing

There is no 100% accurate test for overfitting and there is not a 100% effective way to prevent it. Rather, we may use multiple techniques in combination to prevent overfitting and various methods to detect it.

FH University of Applied Sciences TECHNIKUM WIEN

Spline Regression

Instead of considering a polynomial fitting, use a piecewise function where every piece is a polynomial

$$s : [a, b] \to \mathbb{R}$$

$$s(x) = \begin{cases} s_1(x) \text{ if } x \in [k_0, k_1) \\ s_2(x) \text{ if } x \in [k_1, k_2) \\ \vdots \\ s_K(x) \text{ if } x \in [k_{K-1}, k_K] \end{cases}$$

K knots $\{k_1, ..., k_K\}$

$$a = k_0 < k_1 < \ldots < k_K = b$$

Lower degree polynomials on every piece reduces changes of oscillation

Knot Positioning

Equally spaced between Xmin and Xmax

Equally spaced along the quantiles of the feature

FH University of Applied Sciences TECHNIKUM WIEN

Spline Regression

Truncated Power Basis

$$p(x) = \sum_{j=0}^{p} \beta_j x^j + \sum_{l=1}^{K} \beta_{p+l}(x - k_l)_+^p$$

There are K + p + 1 functions in the truncated power basis for splines

Basis built out of:
- Polynomials (*order p*)

$$h_j(x) = x^j \text{ for } j \in \{0, \ldots, p\}$$

- Truncated power basis

$$h_{p+k}(x) = (x - k_l)_+^p \text{ for } l \in \{1, \ldots, K\}$$

Notation:

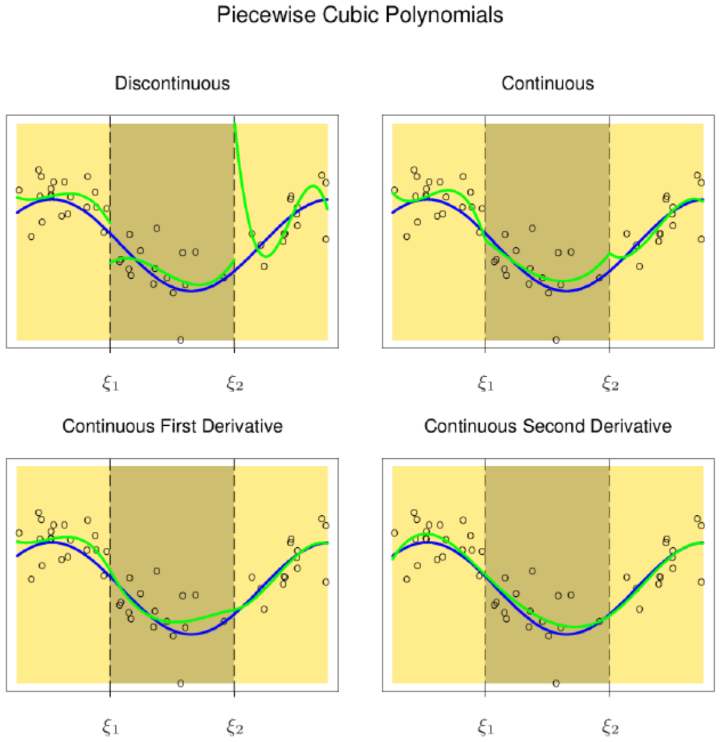
$$x_+^n = \begin{cases} x^n & : x > 0 \\ 0 & : x \leq 0. \end{cases}$$

FH University of Applied Sciences TECHNIKUM WIEN

Spline Regression

Challenges:
- Continuity of the global function
- Smoothness of the function at interface

Smoothing splines use a penalized form of least squares fitting, where penalization is with respect to the 2nd derivative of the estimated functional relationship.



[Image from Hastie et al. (2017): The Elements of Statistical Learning]

Spline Regression

Design Matrix

$$S = \begin{bmatrix} x_1^0 & x_1^1 & x_1^2 & \dots & x_1^d & (x_1 - k_1)_+^d & (x_1 - k_2)_+^d & \dots & (x_1 - k_l)_+^d \\ x_2^0 & x_2^1 & x_2^2 & \dots & x_2^d & (x_2 - k_1)_+^d & (x_2 - k_2)_+^d & \dots & (x_2 - k_l)_+^d \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n^0 & x_n^1 & x_n^2 & \dots & x_n^d & (x_n - k_1)_+^d & (x_n - k_2)_+^d & \dots & (x_n - k_l)_+^d \end{bmatrix}$$

OLS method still holds:

$$\hat{\beta} = (S^T S)^{-1} S^T y.$$

Spline regression can also be seen as a linear optimization problem, like multi-linear regression (interpretation of coefficients is different!)

A spline is characterized by its number of knots and degrees (parameters).

Cubic splines are commonly used, leaving only the number of knots as a parameter

FH University of Applied Sciences TECHNIKUM WIEN

Model Selection

We have seen methods to specify an optimal subset of predictors for a problem.

When a models are nonlinear, a higher degree of complexity is expected.

Cross validation can be used to test a model, in case there is enough data.

FH University of Applied Sciences TECHNIKUM WIEN

The Akaike Information Criterion (AIC)

The same data admits several models: Which one should we use?

- Cross-validation can be used to choose models
- AIC method: select a model that minimizes complexity

$$AIC_l = n \ln\left(\frac{SSR}{n}\right) + 2f$$

$$SSR = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

n = # of observations,
f = degree of freedom of the model

- AIC can be regarded as a maximum-likelihood method

Interpretation: AIC is a loss function that depends both on the predictive error, and the complexity of the model.
We prefer a model with few parameters and low error.

FH University of Applied Sciences TECHNIKUM WIEN

The Akaike Information Criterion (AIC)

- For linear models the AIC takes a particular simple and enlightening form.

$$AIC_l = n \ln\left(\frac{SSR}{n}\right) + 2f$$

$$SSR = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

n = # of observations,
f = degree of freedom of the model

1st term: an increasing function of estimated error
2nd term: penalizes for model complexity.

The AIC is an asymptotically valid quantity:
it needs a large enough sample to use it usually
$n \gg f^2$ by at least an order of magnitude.

Given two linear models with the same number of parameters, the AIC chooses the model with lower SSR.

FH University of Applied Sciences TECHNIKUM WIEN

Degrees of freedom for some linear models

Model	Degree of freedom	Explanation
Simple linear regression $\mathbb{R} \to \mathbb{R}$	3	1 coefficient for the feature, 1 coefficient for the intercept, 1 error term
Multilinear regression $\mathbb{R}^k \to \mathbb{R}$	k+2	k coefficients for features, 1 coefficient for the intercept, 1 error term
Polynomial regression $\mathbb{R} \to \mathbb{R}$ of degree d	d+2	$d+1$ coefficients for the monomial basis, 1 error term
Spline regression $\mathbb{R} \to \mathbb{R}$ of degree d with l nots	d+l+2	$d+l+1$ coefficients for the spline basis, 1 error term

Taka Away

Regression methods permits expanding for modeling nonlinear relations:

- Polynomial regression adds flexibility but risk of overfitting
- Spline regression offers controlled flexility and requires dealing with smoothing

All these methods can be expressed in a linear regression framework

Exercise: One-dimensional regression

(i) Please prepare the exercise as an executable and presentable Python script or notebook.

(ii) You should not use any program libraries that contradict the spirit of the exercise.

1 Setup

Use the Python package `sklearn`.

2 Polynomial regression

Simulate data from a linear function with Gaussian noise. Fit polynomials of varying degrees to this data. Show that overfitting occurs for high degree polynomials. Quantify the fit by the (unadjusted) R^2. Compute the AIC for the models and verify that the AIC does not prefer the model with the best fit.

3 Spline regression

Simulate data from a higher-degree polynomial with Gaussian noise. Fit cubic splines with knots positioned in regular intervals with respect to the quantiles for varying numbers of knots. Quantify the fit by the (unadjusted) R^2. Compute the AIC for the models and choose the number of knots accordingly.