Statistical Learning

Session 5a: Splines and GAMs

Marjolein Fokkema

Leiden University
m.fokkema@fsw.leidenuniv.nl

Splines and GAMs

- Polynomial regression.
- Step functions
- Polynomials and step functions combined: Splines
  - Cubic splines
  - Natural splines
  - Smoothing splines

Modeling non-linearity

- Can use polynomial regression.

- E.g., cubic polynomial: $y_i = \alpha + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \epsilon_i$

- More generally, of degree d (denote order $M = d + 1$):

$$y_i = \alpha + \sum_{j=1}^{d} \beta_j x_i^j + \epsilon_i$$

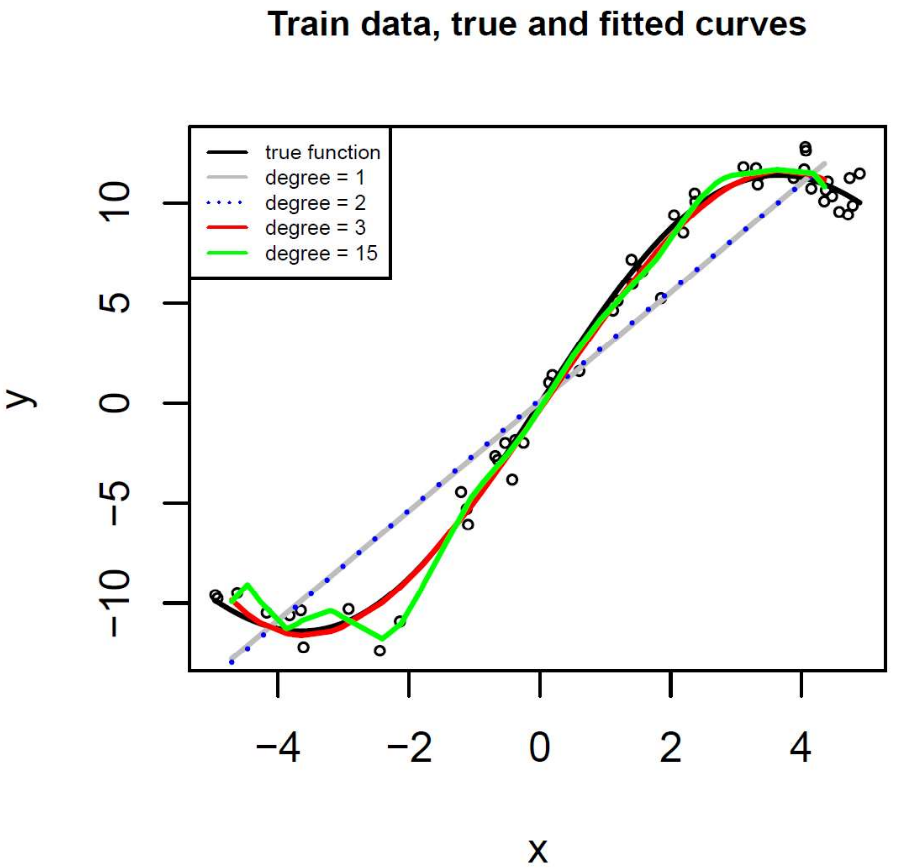
- Higher order polynomials are very flexible functions.

- Can generalize as usual, e.g., for binary outcome:

$$\log\left(\frac{\pi_i}{1 - \pi_i}\right) = \alpha + \sum_{j=1}^{d} \beta_j x_i^j$$

Erratic behavior near boundaries

Exercise 2, session 1:



Worse with lower signal-to-noise ratios, in higher dimensions, with sparser data.

Polynomial Regression

- Interest in approximating shape of association well, no interest in individual coefficients.

- Model selection through:

  1. Statistical tests (ANOVA, likelihood ratio)
  2. Information criteria (AIC, BIC)
  3. Cross validation

- Similar applies for generalized additive models (GAMs).

Step functions

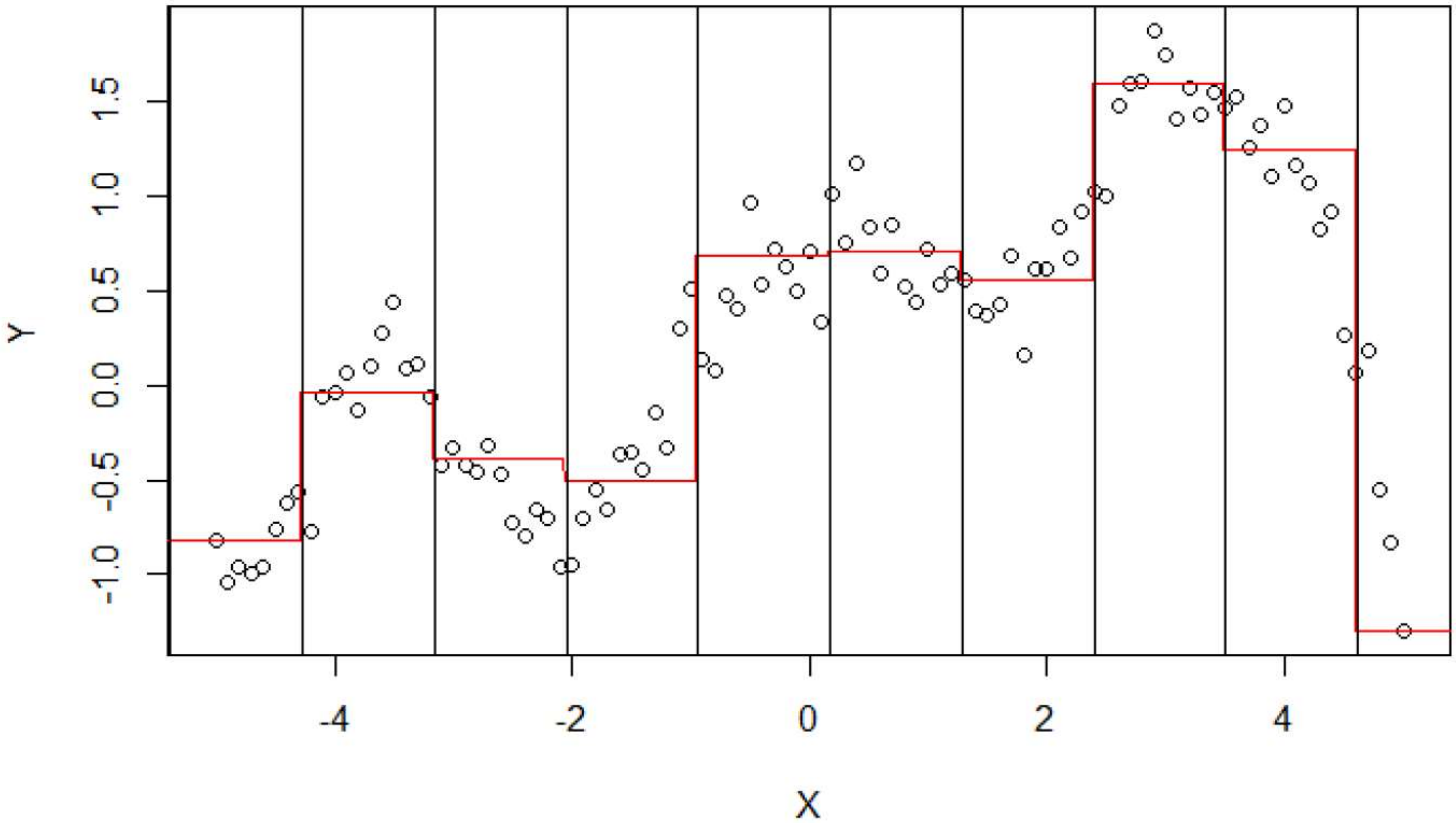
- Define cut points $\xi_1, \xi_2, \xi_3, \ldots, \xi_K$ and with these functions

$$\begin{aligned}
h_0(X) &= I(X < \xi_1) \\
h_1(X) &= I(\xi_1 \leq X < \xi_2) \\
\ldots & \\
h_{K-1}(X) &= I(\xi_{K-1} \leq X < \xi_K) \\
h_K(X) &= I(\xi_K \leq X)
\end{aligned}$$

- Can use $h_k(X)$ as predictors in a regression:

$$y_i = \beta_0 + \sum_{k=1}^{K} \beta_k h_k(x_i) + \epsilon_i$$

Step function



$y_i = \beta_0 + \sum_{k=1}^{K} \beta_k h_k(x_i) + \epsilon_i$

GAM: GLM with flexible basis functions

- Polynomials and step functions are special cases of *basis functions*.
- Define basis functions of each predictor X_j: $h_{j,1}(X_j), h_{j,2}(X_j), \ldots, h_{j,M_j}(X_j)$.
- Use basis functions instead of original X_j as predictors in a (G)LM:

$$f(x_i) = \beta_0 + \sum_{j=1}^{p} \sum_{m_j=1}^{M_j} \beta_{m_j} h_{m_j}(x_{ij}) = \sum_{j=1}^{p} f_j(x_{ij})$$

- Parametric: If functions h_k are fixed and known, coefficients can be estimated 'as usual' (OLS, ML, ...)
- Non-parametric: Smoothing splines, smooth functions are more data driven

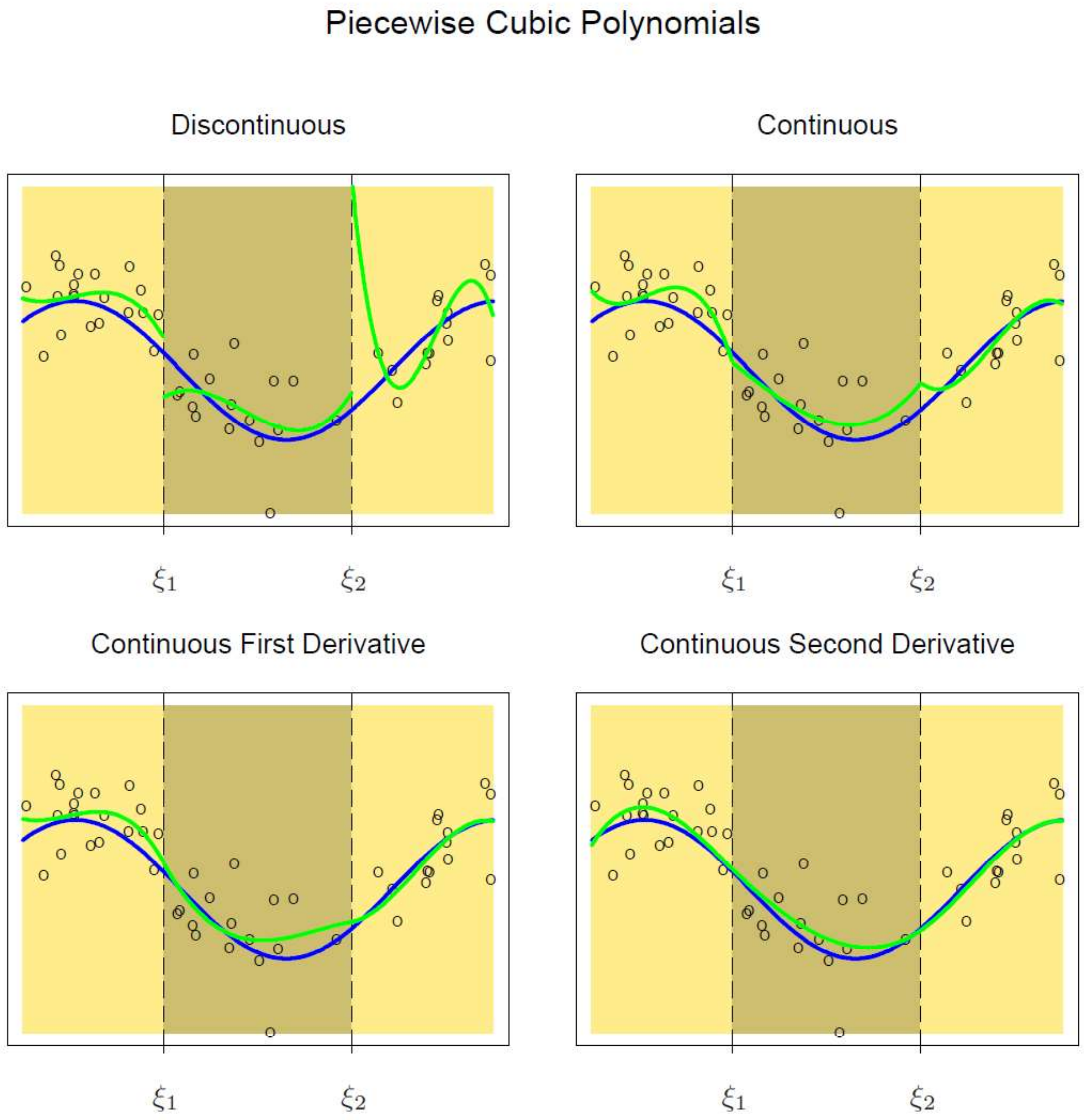
Piecewise polynomial

- Combine step idea with polynomial functions
- Create a cutpoint ξ:

$$\hat{y}_i = \begin{cases} \hat{\beta}_{01} + \hat{\beta}_{11} x_i + \hat{\beta}_{21} x_i^2 + \hat{\beta}_{31} x_i^3 \text{ if } x_i < \xi \\ \\ \hat{\beta}_{02} + \hat{\beta}_{12} x_i + \hat{\beta}_{22} x_i^2 + \hat{\beta}_{32} x_i^3 \text{ if } x_i \geq \xi \end{cases}$$

- This is a *piecewise polynomial* with 1 knot (cutpoint).
- Can give erratic behavior near boundaries:

Piecewise polynomial



Regression Splines

- To obtain a *smooth* curve, we add constraints:
  - f should be continuous.
  - f' should be continuous.
  - f'' should be continuous.

- General definition:
  - A degree-d spline is a piecewise polynomial of degree d,
  - with continuity in the derivatives up to degree $d - 1$.

- It uses $d + 1 + K$ degrees of freedom. A cubic spline with K knots uses $K + 4$ degrees of freedom.

Cubic spline

- The basis functions are given by

$$
\begin{aligned}
h_1(X) &= X \\
h_2(X) &= X^2 \\
h_3(X) &= X^3 \\
h_{3+j}(X, \xi_k) &= (X - \xi_k)_+^3 = \begin{cases} (X - \xi_k)^3 \text{ if} & X > \xi_k \\ 0 & \text{otherwise} \end{cases}
\end{aligned}
$$

  for $k = 1, \ldots, K$

- Note: $(X - \xi_k)_+^3 = (X^3 - 3\xi_k X^2 + 3\xi_k^2 X - \xi_k^3)$.

Natural spline

- Cubic spline often has large variance at the outer ranges of the predictors.
- The *natural spline* introduces additional constraints: the function should be linear at the boundaries.
- Uses lower degrees of freedom: quadratic and cubic effects are zero at both boundaries, yielding $df = K + 2$.

Choosing number and location of knots

- Where should we place the knots?
  - Prior knowledge / information.
  - Place the knots in a uniform way, for example based on quantiles.

- How many knots should we use? (or equivalently: How many degrees of freedom?)
  - Prior knowledge / information.
  - Determine by cross validation.

Smoothing Splines

- Instead of working with *parametric* splines, it is also possible to use the *fit + penalty* approach to fit the smooth functions:

$$PRSS(f, \lambda) = \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \int f''(t)^2 dt$$

- The function f that minimizes the PRSS is known as a *smoothing spline*: A natural spline with a knot at every unique- observed value x_i.

How many basis functions does this yield?

What happens to the 'shape'of f if $\lambda = 0$? And if λ increases?

Smoothing Spline

Use the *fit + penalty* approach to fit the smooth functions:

$$PRSS(f,\lambda) = \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \int f''(t)^2 dt$$

- Thus, $f(x_i)$ has a design matrix $\mathbf{N}$ composed of n columns; it is heavily overparameterized.
- The penalty $\lambda \int f''(t)^2 dt$ shrinks the coefficients of many columns towards zero (c.f. ridge regression).

Generalized Additive Models

- Often we have multiple predictor variables: $X_1, \ldots, X_p$
- We can generalize the ideas:

$$y_i = \beta_0 + \sum_{j=1}^{p} f_j(x_{ij}) + \epsilon_i$$

- Also to other GLM response variable types (e.g., binomial, count).
- The f_j are estimated through minimizing:

$$PRSS = \sum_{i=1}^{N} \{y_i - \beta_0 - \sum_{j=1}^{p} f_j(x_{ij})\}^2 + \sum_{j=1}^{p} \left[\lambda_j \int f_j''(t_j)^2 dt_j\right]$$

- R package `gam` (developed by the authors of the ISL book) uses a backfitting algorithm to estimate models with smoothing splines.

Penalized likelihood estimation

R package `mgcv` (Wood, 2004, 2011) takes a penalized likelihood estimation approach.

- `mgcv` in fact treats smooths as random effects, thus allowing for using REML estimation (specify `method = "REML"` in the call to function `gam`; should generally be preferred over the default GCV estimation approach).
  - The linear (parametric) effect of the smooth is the fixed effect, the non-linear (non-parametric) effects are the random effects: A penalized deviation from the fixed effect.
- `mgcv` works very similar as package `gam`: It provides an estimation function `gam()`, and a function `s()` that is used to specify smooth terms in the model formula.
  - Random intercepts and slopes be included as a smooth function (e.g., `s(subject_id, bs = "re")` would fit a random intercept w.r.t. subjects in a longitudinal analysis).
- `mgcv` is current state-of-the-art, provides enormous flexibility.

References

Wood, S.N. (2004). Stable and efficient multiple smoothing parameter estimation for generalized additive models. *Journal of the American Statistical Association, 99*, 673-686.

Wood, S.N. (2011). Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. *Journal of the Royal Statistical Society (B), 73*(1), 3-36.

Exercise 1

Load the Boston Housing data:

```
library("MASS")
data(Boston)
```

Set up a cubic spline basis for variable `lstat`:

```
library("splines")
basis <- bs(Boston$lstat, df = 5)
```

(a) Where are the knots located? (you can see this if you print `basis`)

(b) How many basis functions were generated?

(c) Create a plot with the value of each basis function on the y-axis and the `lstat` variable on the x-axis. Hint: First reorder the observations in the dataset `Boston <- Boston[order(Boston$lstat), ]` and then use function `matplot`.