STK2100- Machine Learning and Statistical Methods for Prediction and Classification

Spring 2025

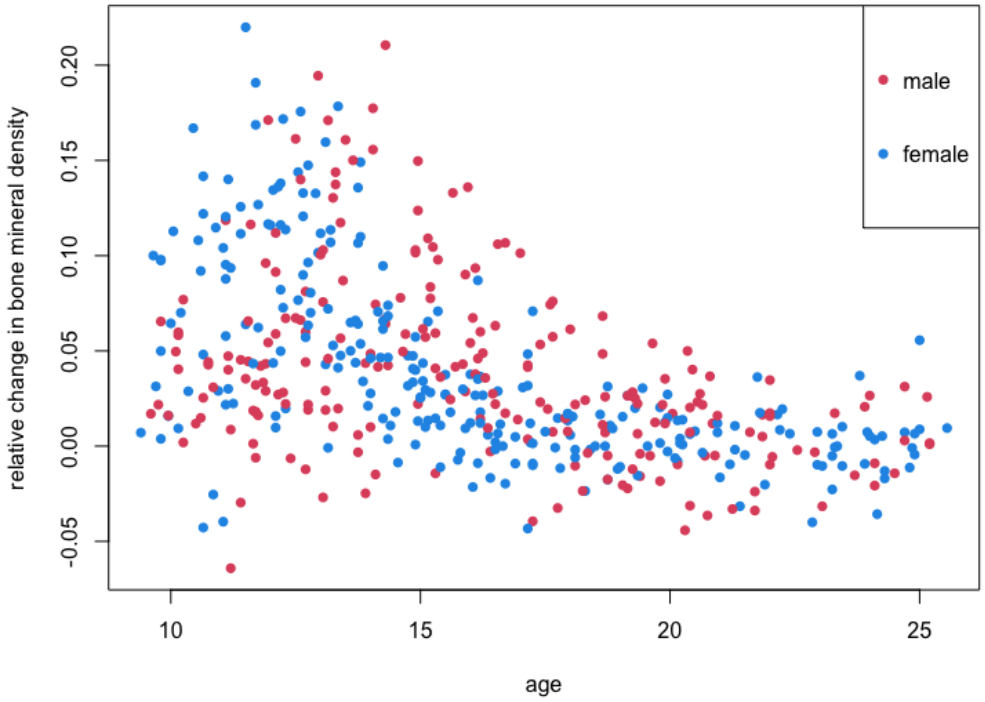
Basis expansions and regularisation

Covers Sections 5.1, 5.2, 5.3, 5.4 (only the first two paragraphs and eq (5.16)), 5.5 and 5.6

Ingrid Hobæk Haff
Department of Mathematics
University of Oslo

Example: Bone mineral density

- Response: Relative change in the bone mineral density at the spine in adolescents (`spbmd`)
- Explanatory variables:
  - Age (`age`)
  - Gender (`gender`)
- We want to predict how fast adolescents grow, here represented by the relative change in bone mineral density in the spine, depending on their age and gender



Going beyond linearity

- When predicting Y based on covariates $\mathbf{x}$, we have so far mostly considered linear models, such as the linear regression model

$$\mathrm{E}(Y|\mathbf{X} = \mathbf{x}) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$$

and the logistic regression model

$$P(Y = 1|\mathbf{X} = \mathbf{x}) = \frac{e^{\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p}}{1 + e^{\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p}}$$

- In many cases, these models not flexible enough, such as for the bone mineral density data, where the effect of age appears to be highly non-linear
- The aim in this chapter is to construct more flexible models based on linear basis expansion in $\mathbf{x}$

$$f(\mathbf{x}) = \sum_{m=1}^{M} \beta_m h_m(\mathbf{x})$$

- Note that given the h_ms, the estimation of $\beta_1, ..., \beta_M$ can be performed as before

Examples of basis expansions

- Examples of basis expansions in $\mathbf{x}$ are
  - Linear regression: $h_m(\mathbf{x}) = x_m, m = 1, ..., p$
  - Projections (for instance principal components): $h_m(\mathbf{x}) = \mathbf{a}_m^T \mathbf{x}$
  - Polynomial terms: $h_m(\mathbf{x}) = x_j^2$ or $h_m(\mathbf{x}) = x_j x_k$
  - Non-linear transformations: $h_m(\mathbf{x}) = \log(x_j), \sqrt{x_j}$, ...
  - Indicators: $h_m(\mathbf{x}) = I(L_m \leq x_k \leq U_m)$
- Polynomials are useful, but limited by their global structure
- An alternative is piecewise polynomials, also called splines

General basis expansions

- The general structure $f(\mathbf{x}) = \sum_{m=1}^{M} \beta_m h_m(\mathbf{x})$ may be too flexible if many transformations allowed
- Hence, we need some control of the complexity, for instance with
  - Restriction: Choose before-hand to limit the class of functions
  - Selection: Start with a large set, and select the most important ones
  - Regularisation: Restrict the β coefficients with ridge or Lasso penalisation

Generalised additive models

- Additive models are given by

$$\mathrm{E}(Y|\mathbf{X}=\mathbf{x}) = f(\mathbf{x}) = \sum_{j=1}^{p} f_j(x_j)$$

- Here, each f_j can be complex:

$$f_j(x_j) = \sum_{m=1}^{M} \beta_{jm} h_{jm}(x_j)$$

- Generalised additive models (GAMs) are given by

$$g(\mathrm{E}(Y|\mathbf{X}=\mathbf{x})) = \sum_{j=1}^{p} f_j(x_j),$$

  for some smooth, invertible link function g
- An example of a GAM is the logistic regression type model

$$\Pr(Y=1|\mathbf{X}=\mathbf{x}) = \frac{\exp(\sum_{j=1}^{p} f_j(x_j))}{1+\exp(\sum_{j=1}^{p} f_j(x_j))}$$

Splines and piecewise polynomials

- For simplicity, we start by considering the one-dimensional case, where x is a scalar
- The aim is to construct $f(x)$ by dividing the domain into intervals, with a separate polynomial within each interval
- Let us first divide the domain into three intervals: $x \leq \xi_1$, $\xi_1 < x \leq \xi_2$ and $x > \xi_2$
- The simplest model is to use piecewise constants, i.e. polynomials of order 0

$$h_1(x) = I(x \leq \xi_1), \quad h_2(x) = I(\xi_1 < x \leq \xi_2), \quad h_3(x) = I(\xi_2 < x)$$

- The least squares estimate of the model is then $\hat{\beta}_m = \bar{y}_m$, $m = 1, 2, 3$, i.e. the mean of the ys in each of the three regions defined by ξ_1 and ξ_2

Splines and piecewise polynomials

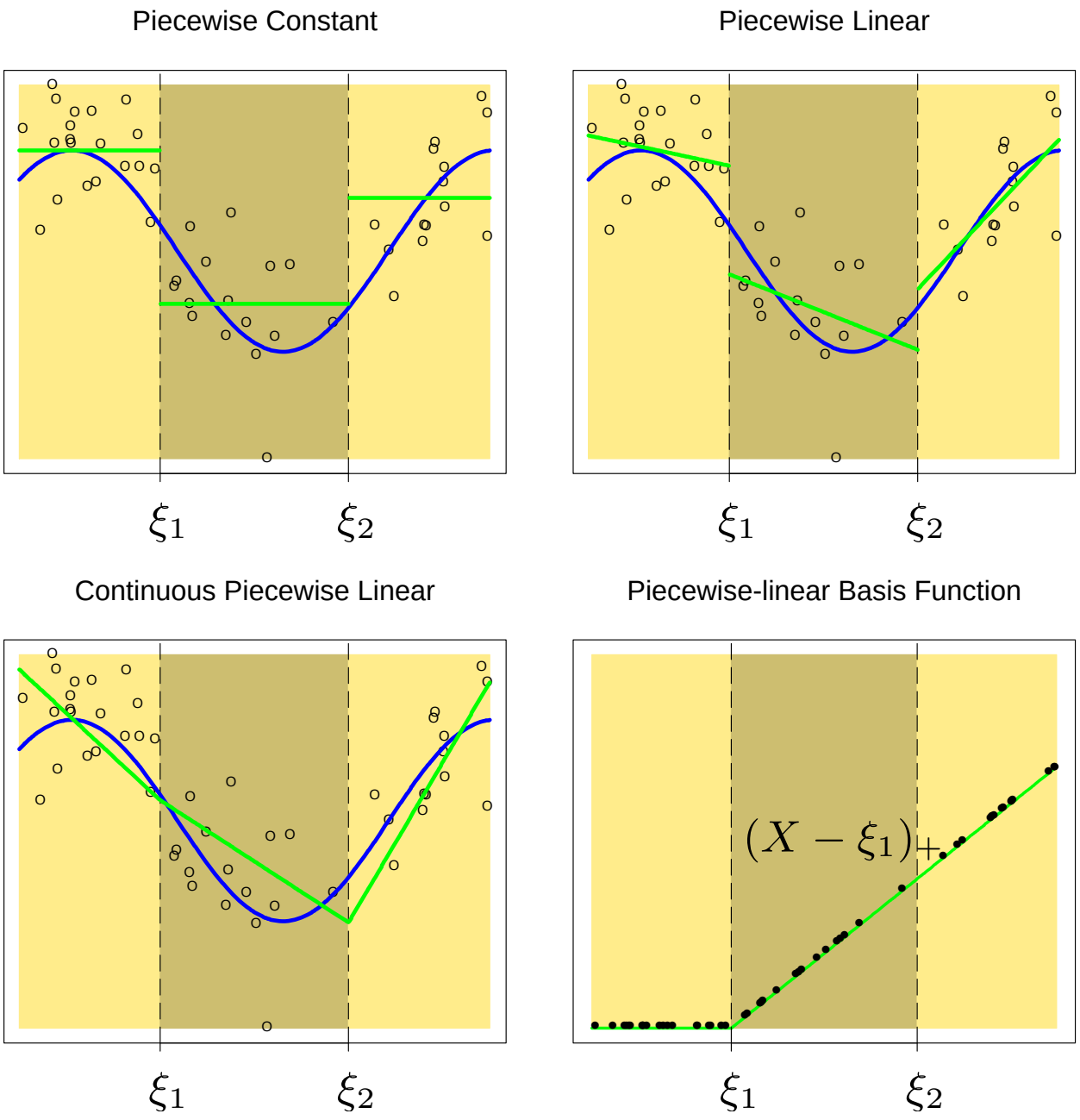
- The model can be made more flexible by extending to piecewise linear functions, i.e. polynomials of order 1
- This requires the definition of three extra basis functions given by

$$h_{m+3}(x) = h_m(x) \cdot x, \quad m = 1, 2, 3,$$

  where $h_1(x)$, $h_2(x)$ and $h_3(x)$ are as before
- However, this gives a solution that is discontinuous at ξ_1 and ξ_2, and we would typically prefer a continuous one
- Imposing continuity, we reduce from 6 to 4 basis functions:

$$h_1(x) = 1, \quad h_2(x) = x, \quad h_3(x) = (x - \xi_1)_+, \quad h_4(x) = (x - \xi_2)_+$$

Splines and piecewise polynomials – illustration



General definition of splines

- Extending to piecewise quadratic with continuity constraints on $f(x)$ and $f'(x)$, we define the 5 basis functions

$$\begin{aligned} h_1(x) =& 1, \quad h_2(x) = x, \quad h_3(x) = x^2 \\ h_4(x) =& (x - \xi_1)_+^2, \quad h_5(x) = (x - \xi_2)_+^2 \end{aligned}$$

- More generally, an order-M spline with knots $\xi_1 < \xi_2 < \cdots < \xi_K$ is a piecewise polynomial of order $M - 1$ with continuous derivatives up to order $M - 2$
- Such a spline is given by

$$\begin{aligned} h_j(x) =& x^{j-1}, & j = 1, ..., M \\ h_{M+l}(x) =& (x - \xi_l)_+^{M-1}, & l = 1, ..., K \end{aligned}$$

General definition of splines

- Cubic splines correspond to $M = 4$, and are claimed to be the lowest-order splines for which the knot-discontinuity is not visible to the human eye
- Splines with fixed knots are also known as regression splines
- In practice, one needs to specify
  - the number of knots, K
  - the placements of the knots, $\xi_l, l = 1, ..., K$
  - the order of spline, M
- For cubic splines, there are
  - $4(K + 1)$ parameters ($K + 1$ intervals)
  - $3K$ constraints (continuity of $f(x)$, $f'(x)$ and $f''(x)$ at each of the K knots)
  - thus, $4(K + 1) - 3K = K + 4$ free parameters

Natural cubic splines

- Cubic splines are generally given by

$$f(x) = \beta_1 + \beta_2 x + \beta_3 x^2 + \beta_4 x^3 + \sum_{l=1}^{K} \beta_{4+l}(x - \xi_l)_+^3$$

- However, the variability in the two end intervals $(-\infty, \xi_1)$ and (ξ_K, ∞) is huge
- In order to reduce this variability, at the cost of higher bias, it is usual to impose linearity in the first and last intervals, i.e.

$$f''(\xi_1) = f''(\xi_K) = f^{(3)}(\xi_1) = f^{(3)}(\xi_K) = 0$$

- Cubic splines with these additional constaints are called natural cubic splines
- The 4 degrees of freedom, that are "freed" by imposing these constraints, can instead be used to increase the number of knots

Natural cubic splines

- For $x < \xi_1$ we have

$$f(x) = \beta_1 + \beta_2 x + \beta_3 x^2 + \beta_4 x^3,$$

so that the constraints are

$$f''(\xi_1) = 2\beta_3 + 6\beta_4\xi_1 = 0 \quad \text{and} \quad f^{(3)}(\xi_1) = 6\beta_4 = 0,$$

which leads to

$$\beta_3 = \beta_4 = 0$$

- For $x \geq \xi_K$, we have

$$f(x) = \beta_1 + \beta_2 x + \sum_{j=1}^{K} \beta_{4+j}(x - \xi_j)^3,$$

which gives the constraints

$$\begin{aligned} f''(\xi_K) =& 6\sum_{l=1}^{K} \beta_{4+l}(\xi_K - \xi_l) = 6\sum_{l=1}^{K-1} \beta_{4+l}(\xi_K - \xi_l) = 0 \\ f^{(3)}(\xi_K) =& 6\sum_{l=1}^{K} \beta_{4+l} = 0 \end{aligned}$$

Natural cubic splines

- The first constraint implies

$$\begin{aligned}
\beta_{3+K}(\xi_K - \xi_{K-1}) &= -\sum_{l=1}^{K-2} \beta_{4+l}(\xi_K - \xi_l) \\
\iff \beta_{3+K} &= -\sum_{l=1}^{K-2} \beta_{4+l} \frac{\xi_K - \xi_l}{\xi_K - \xi_{K-1}}
\end{aligned}$$

- The second constraint implies

$$\begin{aligned}
\beta_{4+K} = -\sum_{j=1}^{K-1} \beta_{4+j} &= -\beta_{3+K} - \sum_{j=1}^{K-2} \beta_{4+j} \\
&= \sum_{l=1}^{K-2} \beta_{4+l} \left(\frac{\xi_K - \xi_l}{\xi_K - \xi_{K-1}} - 1 \right) \\
&= \sum_{l=1}^{K-2} \beta_{4+l} \frac{\xi_{K-1} - \xi_l}{\xi_K - \xi_{K-1}}
\end{aligned}$$

Regression splines

- Inserting the results, we can re-write $f(x)$ as

$$f(x) = \beta_1 + \beta_2 x + \sum_{j=1}^{K-2} \tilde{\beta}_{j+2} N_{j+2}(x)$$

with

$$\begin{aligned} \tilde{\beta}_{j+2} &= \beta_{4+j}(\xi_K - \xi_j) \\ N_{j+2}(x) &= d_k(x) - d_{K-1}(x) \\ d_k(x) &= \frac{(x-\xi_k)_+^3 - (x-\xi_K)_+^3}{\xi_K - \xi_k} \end{aligned}$$

- $f(x)$ is then parametrised by a spline, and is a linear combination of basis functions $N_j(x)$
- The $N_j(x)$s can be seen as derived covariates
- Non linear pre-processing of covariates can also be done, and is relevant for instance in image and signal recognition

Regression splines

- Given the N_js, we have a standard regression (or classification) problem with $\mathbf{x}_i = (1, x_i, N_3(x_i), ..., N_K(x_i))^T$
- For additive models $Y = f(x) + \varepsilon$, $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_K)^T$ is estimated by least squares

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

- For GAMs, such as the one on logistic form, $\boldsymbol{\beta}$ can be estimated by maximum likelihood, as we will see later
- One needs to specify K, which is a tuning parameter, and the knots $\xi_1 < \xi_2 < \cdots < \xi_K$
- The knots are typically chosen either uniformly along the x-axis, or as the $1/(K+1) \cdot 100\%, \ldots, K/(K+1) \cdot 100\%$ percentiles of the observations $x_1, \ldots, x_N$
- This can be expanded to several covariates, i.e. $p > 1$

Smoothing splines for additive models

- A penalised least squares criterion is given by

$$RSS(f, \lambda) = \sum_{i=1}^{n} [y_i - f(x_i)]^2 + \lambda \int_{-\infty}^{\infty} [f''(t)]^2 dt, \quad \lambda > 0$$

- This penalises the roughness of f in order to obtain a smoother solution
- For $\lambda = 0$, one obtains a perfect fit at (x_i, y_i)
- For $\lambda = \infty$, we must have $f''(x) = 0$, so that $f(x)$ is linear
- It can be shown that the minimiser of $RSS(f, \lambda)$ is a natural cubic spline with knots $\xi_i = x_i, i = 1, ..., N$, i.e.

$$f(x) = \sum_{j=1}^{N} \theta_j N_j(x)$$

Smoothing splines for additive models

- We can now re-write the criterion as a generalised ridge regression

$$RSS(f, \lambda) = (\mathbf{y} - \mathbf{N}\boldsymbol{\theta})^T(\mathbf{y} - \mathbf{N}\boldsymbol{\theta}) + \lambda\boldsymbol{\theta}^T\boldsymbol{\Omega}\boldsymbol{\theta}$$

  where $\mathbf{N}$ is the matrix with elements $N_{ij} = N_j(x_i)$ and $\boldsymbol{\Omega}$ is the matrix with elements $\Omega_{jk} = \int N_j''(t)N_k''(t)dt$
- Hence, we have

$$\begin{aligned} \frac{\partial RSS(f, \lambda)}{\partial \boldsymbol{\theta}} &= -2\mathbf{N}^T\mathbf{y} + 2\mathbf{N}^T\mathbf{N}\boldsymbol{\theta} + 2\lambda\boldsymbol{\Omega}\boldsymbol{\theta} \\ &= -2\mathbf{N}^T\mathbf{y} + 2(\mathbf{N}^T\mathbf{N} + \lambda\boldsymbol{\Omega})\boldsymbol{\theta} \end{aligned}$$

  with the solution

$$\hat{\boldsymbol{\theta}} = (\mathbf{N}^T\mathbf{N} + \lambda\boldsymbol{\Omega})^{-1}\mathbf{N}^T\mathbf{y}$$

- The penalisation of the roughness of f becomes a penalisation of the size of the coefficients θ_j

Degrees of freedom and smoother matrices

- The prediction is then given by

$$\hat{\mathbf{y}} = \widehat{f}(x) = \mathbf{N}\hat{\theta} = \underbrace{\mathbf{N}(\mathbf{N}^T\mathbf{N} + \lambda\Omega)^{-1}\mathbf{N}^T}_{\mathbf{S}_\lambda}\mathbf{y} = \mathbf{S}_\lambda\mathbf{y},$$

  and is thus linear in $\mathbf{y}$
- $\mathbf{S}_\lambda$ is called a smoother matrix
- A common way to define the complexity or degrees of freedom of $\widehat{f}(x)$ is $df = \mathrm{tr}(\mathbf{S}_\lambda)$
- We can use the one-to-one correspondence between df and λ to specify λ
- Using `smooth.spline(x,y,df)`, one can try out different dfs and select according to some criterion, for instance F-tests or residual plots

Specification of λ by cross-validation

- An alternative is to choose λ with cross-validation
- The integrated expected prediction error is given by

$$EPE(\widehat{f}_\lambda) = \mathrm{E}\left((Y - \widehat{f}_\lambda(X))^2\right) = \sigma_\varepsilon^2 + \mathrm{E}\left((f(X) - \widehat{f}_\lambda(X))^2\right) = \sigma_\varepsilon^2 + MSE(\widehat{f}_\lambda)$$

with $MSE(\widehat{f}_\lambda) = \mathrm{E}\left((Bias(\widehat{f}_\lambda)^2\right) + \mathrm{E}\left(\mathrm{Var}(\widehat{f}_\lambda)\right)$

- Thus, the choice of λ is a tradeoff between bias and variance
- The mean squared error can be approximated by cross-validation
- It can be shown that the LOOCV criterion can be written as

$$\mathrm{CV}(\widehat{f}_\lambda) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \widehat{f}_\lambda^{-i}(x_i))^2 = \frac{1}{N}\sum_{i=1}^{N}\left((y_i - \widehat{f}_\lambda(x_i))/(1 - S_\lambda(i,i))\right)^2$$

- Hence, we only need to fit $\widehat{f}_\lambda$ once per λ-value

Example: Bone mineral density

See R code

Logistic regression and smoothing splines

- Assume that $Y \in \{0, 1\}$ and $\mathbf{x} \in \mathbb{R}^p$
- The ordinary logistic regression can be extended with smoothing splines

$$\Pr(Y = 1|\mathbf{x}) = \frac{\exp(f(\mathbf{x}))}{1 + \exp(f(\mathbf{x}))} \quad \text{with} \quad f(\mathbf{x}) = \sum_{j=1}^{p} f_j(x_j)$$

- For the observations $(\mathbf{x}_i, y_i)$, $i = 1, ..., N$, the log-likelihod function is given by

$$l(f) = \sum_{i=1}^{N} (y_i f(\mathbf{x}_i) - \log(1 + \exp(f(\mathbf{x}_i)))$$

- A penalised log-likelihood criterion is given by

$$l(f; \lambda) = \sum_{i=1}^{N} (y_i f(\mathbf{x}_i) - \log(1 + \exp(f(\mathbf{x}_i))) - \frac{1}{2}\lambda \sum_{j=1}^{p} \int (f_j''(t))^2 dt$$

Logistic regression and smoothing splines

- As for regression, it can be shown that the optimal f_js are natural spline with knots at the unique values of $\{x_{1,j}, ..., x_{N,j}\}$, i.e.

$$f_j(x_j) = \sum_{k=1}^{N} N_{jk}(x_j)\theta_{jk},$$

  where the $N_{jk}(x_j)$s are third order polynomials
- For $p = 1$, letting $\mathbf{N}(x_i) = (N_1(x_i), \ldots, N_N(x_i))^T$, we get

$$\begin{aligned} l(f;\lambda) &= \sum_{i=1}^{N}\left(y_i \sum_{k=1}^{N}\theta_k N_k(x_j) - \log\left(1 + \exp(\sum_{k=1}^{N}\theta_k N_k(x_j))\right)\right) \\ &\quad -\frac{1}{2}\lambda \int \left(\sum_{j=1}^{N} N_j''(t)\theta_j\right)^2 dt \\ &= \sum_{i=1}^{N}\left(y_i \mathbf{N}(x_i)^T\boldsymbol{\theta} - \log\left(1 + \exp(\mathbf{N}(x_i)^T\boldsymbol{\theta})\right)\right) - \frac{1}{2}\lambda\boldsymbol{\theta}^T\boldsymbol{\Omega}\boldsymbol{\theta} \end{aligned}$$

Logistic regression and smoothing splines

- We have

$$\begin{aligned}\frac{\partial l(f;\lambda)}{\partial \boldsymbol{\theta}} &= \sum_{i=1}^{N}\left(y_i\mathbf{N}(x_i) - \frac{\exp(\mathbf{N}(x_i)^T\boldsymbol{\theta})}{1+\exp(\mathbf{N}(x_i)^T\boldsymbol{\theta})}\mathbf{N}(x_i)\right) - \lambda\boldsymbol{\Omega}\boldsymbol{\theta} \\ &= \mathbf{N}^T(\mathbf{y}-\mathbf{p}) - \lambda\boldsymbol{\Omega}\boldsymbol{\theta}\end{aligned}$$

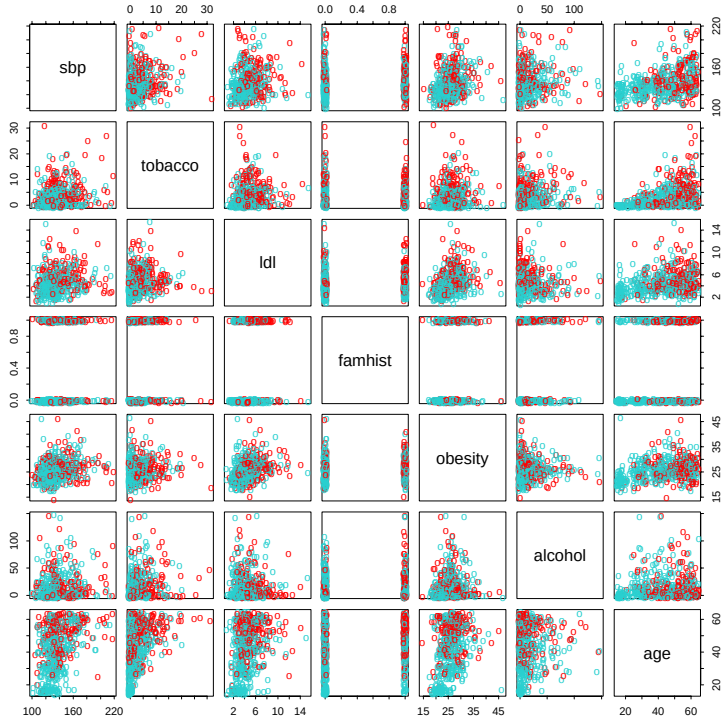
  with $\mathbf{p} = (p_1, \ldots, p_N)^T$, $p_i = P(Y_i = 1|x_i)$, $i = 1, \ldots, N$
- Further, we have

$$\frac{\partial^2 l(f;\lambda)}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^T} = -\mathbf{N}^T\mathbf{W}\mathbf{N} - \lambda\boldsymbol{\Omega},$$

  where $\mathbf{W}$ is the diagonal matrix with diagonal entries $p_i(1-p_i)$
- The log-likelihood is optimised numerically with the Newton-Raphson algorithm

Example: South African Heart Disease

- Response: Presence or absence of myocardial infarction (CHD) Yes/No (`chd`)
- Explanatory variables:
  - Systolic blood pressure (`sbp`)
  - Total lifetime usage of tobacco in kg (`tobacco`)
  - LDL cholesterol (`ldl`)
  - Family history of CHD Yes/No (`famhist`)
  - BMI (`obesity`)
  - Alcohol consumption (`alcohol`)
  - Age (`age`)



See R code