Chapter 7

Flexible regression and classification methods

7.1 Introduction

The curse of dimensionality makes it virtually impossible to estimate a regression function $m(x) = \mathbb{E}[Y|X = x]$ or the probability functions $\pi_j(x) = \mathbb{P}[Y = j|X = x]$ (for classification) in a full nonparametric way without making some structural assumptions.

We are going to describe some flexible models and methods which are of nonparametric nature but making some structural assumptions. In the sequel, we will denote either a regression function $\mathbb{E}[Y|X = x]$ or the logit transform in a binary classification problem $\log(\pi(x)/(1 - \pi(x))$ by

$$g(\cdot) : \mathbb{R}^p \to \mathbb{R}.$$

7.2 Additive models

Additive models require additivity of the function: the model is

$$\begin{aligned} g_{add}(x) &= \mu + \sum_{j=1}^{p} g_j(x_j),\ \ \mu \in \mathbb{R} \\ g_j(\cdot) &: \mathbb{R} \to \mathbb{R}, \quad \mathbb{E}[g_j(X_j)] = 0 \ \ (j = 1, \ldots, p). \end{aligned}$$

The functions $g_j(\cdot)$ are fully nonparametric. The requirement about $\mathbb{E}[g_j(X_j)] = 0$ yields an identifiable model; note that otherwise we could add and subtract constants into the one-dimensional functions $g_j(\cdot)$.

Additive models are generalizations of linear models. While the additive functions are very general, they do not allow for interaction terms such as $g_{j,k}(x_j, x_k)$[1]. The curse of dimensionality is *not* present in additive models. When assuming continuous second derivatives for all the functions g_j, it can be shown that some estimators $\hat{g}_{add}(\cdot)$ have mean square error rate $O(n^{-4/5})$ as for estimating a single one-dimensional ($\mathbb{R} \to \mathbb{R}$) function.

[1]as a matter of fact, both the original Hastie-Tibshirani and the new algorithm in `mgcv` allow for bivariate interactions, optionally

7.2.1 Backfitting for additive regression models

Backfitting is a very general tool for estimation in additive (and other) structures. We can use "any" nonparametric smoothing technique for estimating one-dimensional functions. A smoother against the predictor variables $X_{1j}, \ldots X_{nj}$ is denoted by the hat operator

$$\mathcal{S}_j : (U_1, \ldots, U_n)^\intercal \mapsto (\hat{U}_1, \ldots, \hat{U}_n)^\intercal \ (j = 1, \ldots p)$$

for any response vector $(U_1, \ldots, U_n)^\intercal$. The subscript j indicates that smoothing is done against the jth predictor variable. For example, $\mathcal{S}_j$ could be smoothing splines with the same degrees of freedom for all j, e.g. equal to 5 or estimated by (generalized) cross-validation. Or they could be Nadaraya-Watson Gaussian kernel estimators with the same bandwidth which may be estimated by (generalized) cross-validation.

Backfitting for additive regression then works as follows.

1. Use $\hat{\mu} = n^{-1} \sum_{i=1}^{n} Y_i$. Start with $\hat{g}_j(\cdot) \equiv 0$ for all $j = 1, \ldots, p$.

2. Cycle through the indices $j = 1, 2, \ldots, p, 1, 2, \ldots, p, 1, 2, \ldots$ while computing

$$\hat{\mathbf{g}}_j = \mathcal{S}_j(\mathbf{Y} - \hat{\mu}\mathbf{1} - \sum_{k \neq j} \hat{\mathbf{g}}_k),$$

   where $\mathbf{Y} = (Y_1, \ldots, Y_n)^\intercal$ and $\hat{\mathbf{g}}_j = (\hat{g}_j(X_{1j}), \ldots, \hat{g}_j(X_{nj}))^\intercal$. Stop the iterations if the individual functions $\hat{g}_j(\cdot)$ do not change much anymore, e.g.,

$$\frac{\|\hat{\mathbf{g}}_{j,new} - \hat{\mathbf{g}}_{j,old}\|_2}{\|\hat{\mathbf{g}}_{j,old}\|_2} \leq \text{tol}$$

   where tol is a tolerance such as 10^{-6}.

3. Normalize the functions

$$\tilde{g}_j(\cdot) = \hat{g}_j(\cdot) - n^{-1} \sum_{i=1}^{n} \hat{g}_j(X_{ij}).$$

We may view backfitting as a method to optimize a high-dimensional (penalized) parametric problem. For example, with smoothing spline smoothers, we have seen in chapter 3 that the smoothing spline fit can be represented in terms of basis functions and we have to solve a penalized parametric problem. Backfitting can then be viewed as a coordinate-wise optimization method which optimizes one coordinate (corresponding to one predictor variable) at a time while keeping all others (corresponding to all other predictors) fixed. This coordinate-wise optimization may be slow but backfitting is a very general overall method which directly allows to use one-dimensional smoothing algorithms.

7.2.2 Additive model fitting in R

Additive models can be fitted in R with the function `gam` (**g**eneralized **a**dditive **m**odel) from package `mgcv`. The term "generalized" allows also to fit additive logistic regression, among other things.

The function `gam` uses for the smoothers $\mathcal{S}_j$ a penalized regression spline: i.e., a spline with selected knots including a penalty term (this is somewhat different than a smoothing spline). Interestingly, the function will choose the degrees of freedom, which may be different for every fitted function $\hat{g}_j(\cdot)$, via generalized cross-validation: in particular, this allows to use more degrees of freedom for "complex" functions and few degrees of freedom for functions which seem "simple".

We consider as an example the daily ozone concentration in the Los Angeles basin as a function of 9 predictor variables. The commands and output in R look as follows.

```
library(mgcv)
data(ozone, package = "gss")
d.ozone ← ozone; colnames(d.ozone)[c(1,2,3,4)] ← ..... # ''better'' names
pairs(d.ozone, pch = ".", gap = 0.1) #  --> Scatterplot Matrix
```

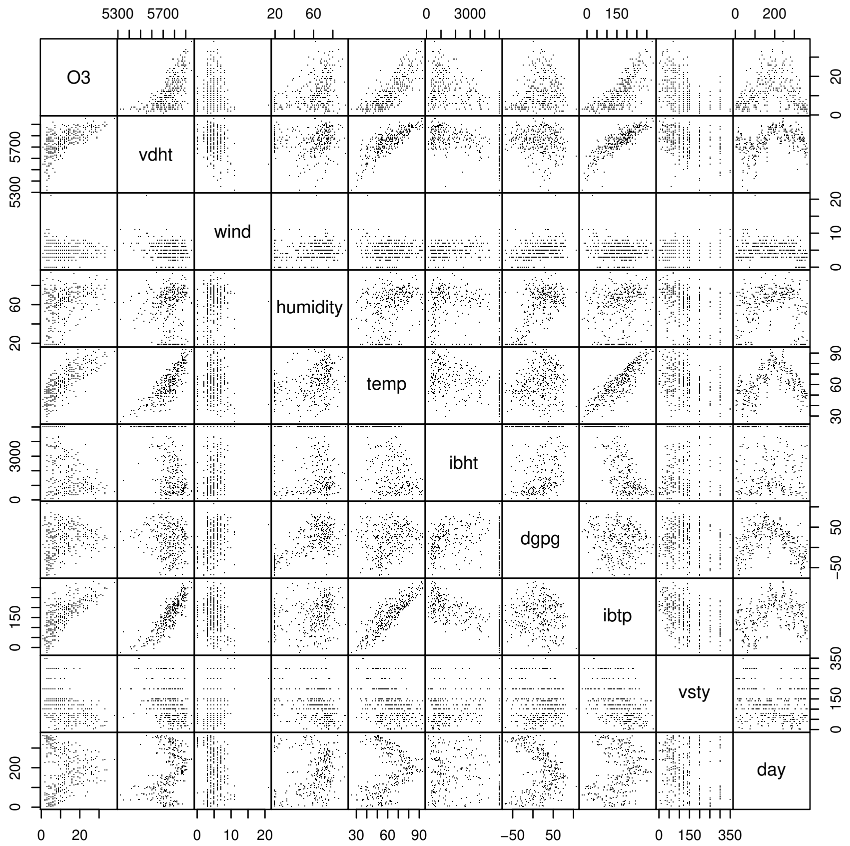

Figure 7.1: Daily ozone concentration in the Los Angeles basin as a function of 9 predictor variables.

```
fit ←
 gam(O3 ~ s(vdht)+s(wind)+s(humidity)+s(temp)+s(ibht)+s(dgpg)+s(ibtp)+s(vsty)+s(day),
     data = d.ozone)
summary(fit)


         . . . . . . . .

    Parametric coefficients:
                Estimate Std. Error t value Pr(>|t|)
    (Intercept)  11.7758     0.1988   59.22   <2e-16 ***
```

```
Approximate significance of smooth terms:
               edf Est.rank      F  p-value
s(vdht)      1.000        1  9.360  0.00242 **
s(wind)      1.000        1  4.375  0.03729 *
s(humidity) 3.631        8  2.595  0.00933 **
s(temp)      4.361        9  4.694 7.56e-06 ***
s(ibht)      3.043        7  1.708  0.10658
s(dgpg)      3.230        7  7.916 7.94e-09 ***
s(ibtp)      1.939        4  1.809  0.12698
s(vsty)      2.232        5  3.825  0.00225 **
s(day)       4.021        9 10.174 1.04e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.797   Deviance explained = 81.2%
GCV score = 14.137   Scale est. = 13.046    n = 330
```

The column `edf` shows the estimated (via GCV) degrees of freedom: if they are equal to 1, a linear straight line is fitted. We get an indication about the relevance of a predictor

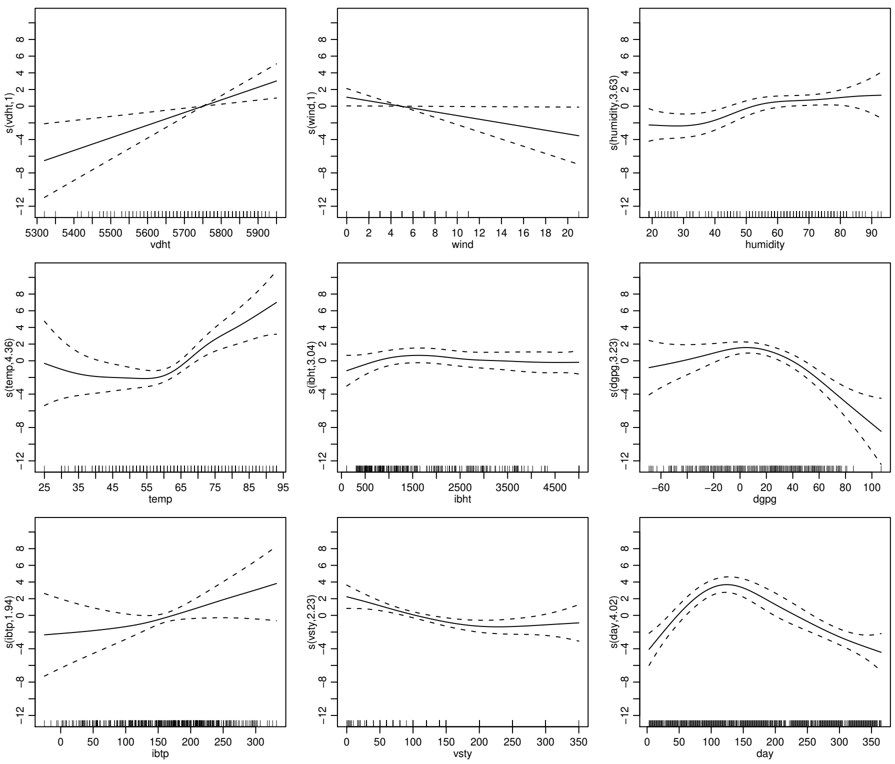

Figure 7.2: Estimated function $\hat{g}_j$ for the ozone data set.

either from Figure 7.2 or from the P-values in the summary output from R .

The fit of the model can be checked via residual analysis as in chapter 1. The Tukey-Anscombe plot indicates heteroscedastic errors. We thus try the log-transform for the response variable and re-do the whole analysis. The results are given below.

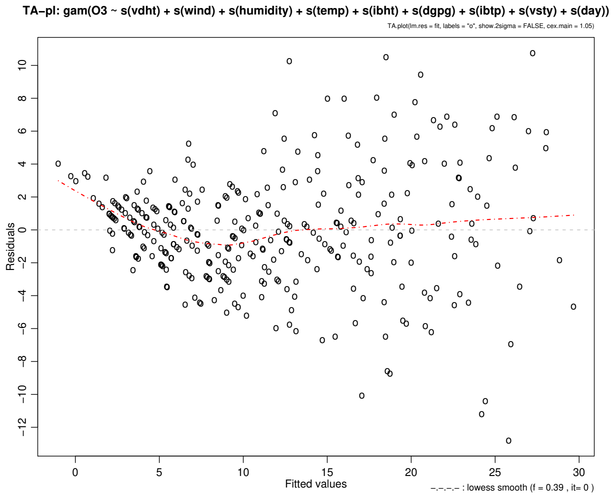

Figure 7.3: Tukey-Anscombe plot for the ozone data set.

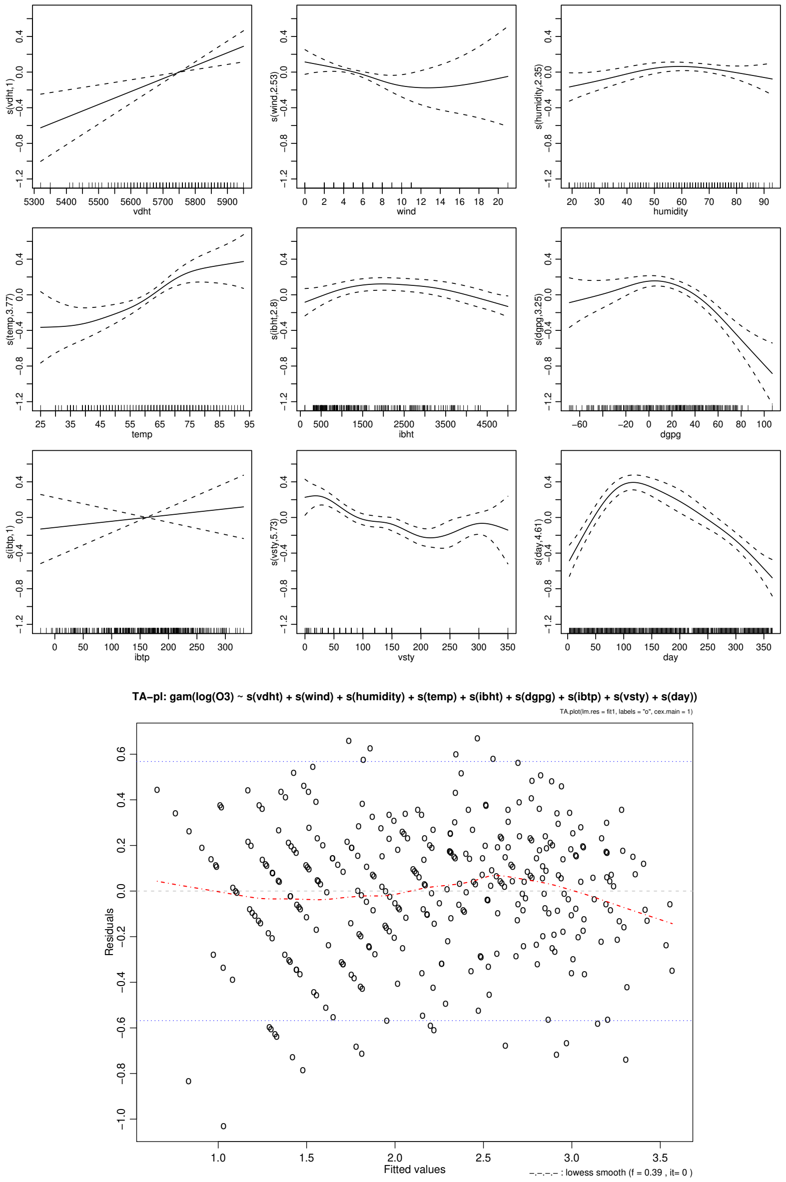

Figure 7.4: Estimated additive functions and Tukey-Anscombe plot for the ozone dataset with `log(O3)` as the response variable.

7.3 MARS

MARS is a shortcut for **m**ultivariate **a**daptive **r**egression **s**plines. MARS is an adaptive procedure for regression and it is often useful for high-dimensional problems with many predictor variables.

MARS uses expansions in piecewise linear basis functions of the form

$$(x_j - d)_+ = \begin{cases} x_j - d & \text{if } x_j > d, \\ 0 & \text{otherwise.} \end{cases}$$

and $(d - x_j)_+$. The value d is a knot and x_j ($j \in \{1, \dots, p\}$) is the j-th component of $x \in \mathbb{R}^p$. The pair $(x_j - d)_+$, $(d - x_j)_+$ is called a *reflected pair*. Note that we use $(x_j - d)_+$ as a function of x ($\in \mathbb{R}^p$) where only the jth component of x is relevant. The collection of reflected pairs is then a set of basis functions

$$\mathcal{B} = \{(x_j - d)_+, (d - x_j)_+; \qquad d \in \{x_{1,j}, x_{2,j}, \dots, x_{n,j}\},\ j \in \{1, \dots, p\}\},$$

where we note that in the univariate case ($p = 1$), $\mathcal{B}$ spans the space of continuous piecewise linear functions, i.e., linear splines.

MARS employs now a forward selection of reflected pairs of basis functions in $\mathcal{B}$ and their products. The model has the form

$$g(x) = \mu + \sum_{m=1}^{M} \beta_m h_m(x),$$

where each function $h_m(\cdot)$ is a function from the basis $\mathcal{B}$ or a product of functions from $\mathcal{B}$. The model building technique is going in a forward way as follows:

1. Start with the function $h_0(x) \equiv 1$. Initialize the model set $\mathcal{M} = \{h_0(\cdot) \equiv 1\}$. Fit the function h_0 by least squares regression, yielding the estimate $\hat{\mu} = n^{-1} \sum_{i=1}^{n} Y_i$.

2. For $r = 1, 2, \dots$ do the following:
   Search for the best pair of functions $((h_{2r-1}(\cdot), h_{2r}(\cdot))$ which are of the form

   $$\begin{aligned} h_{2r-1}(\cdot) &= h_\ell(\cdot) \times (x_j - d)_+, \\ h_{2r}(\cdot) &= h_\ell(\cdot) \times (d - x_j)_+, \end{aligned} \tag{7.1}$$

   for some h_ℓ in the model set $\mathcal{M}$, and some basis functions in $\mathcal{B}$. The best pair of functions is defined to be the one which reduces residual sum of squares most.
   The model fit is then

   $$\hat{g}(x) = \hat{\mu} + \sum_{m=1}^{2r} \hat{\beta}_m h_m(x),$$

   where the coefficients $\hat{\beta}_m$ are estimated by least squares.
   Enlarge the model set in every iteration (with index r) by

   $$\mathcal{M} = \mathcal{M}_{old} \cup \{h_{2r-1}(\cdot), h_{2r}(\cdot)\},$$

   with the functions h_{2r-1}, h_{2r} from (7.1).

3. Iterate step 2 until a large enough number of basis functions $h_m(\cdot)$ has been fitted.

4. Do backward deletion of those pairs of functions $h_{2r-1}(\cdot), h_{2r}(\cdot)$ which increase the residual sum of squares the least.