Additive models

Data Mining - CdL CLAMSES

Tommaso Rigon
Università degli Studi di Milano-Bicocca

Homepage

The Great Barrier Reef

- In this unit we will cover the following **topics**:
  - Generalized additive models (GAMs)
  - Multivariate Adaptive Regression Splines (MARS)
- We have seen that **fully nonparametric** methods are plagued by the **curse of dimensionality**.
- GAMs and MARS are **semi-parametric** approaches that keep the model complexity under control so that:
  - they are more flexible than linear models;
  - they are not hugely impacted by the curse of dimensionality.
- The running example is about **trawl data** from the **Great Barrier Reef**.

An ecological application

The `trawl` dataset

- We consider the `trawl` dataset, which refers to a **survey** of the **fauna** on the sea bed lying between the coast of northern Queensland and the **Great Barrier Reef**.
- The **response** variable is `Score`, which is a standardized numeric quantity measuring the amount of fishes caught on a given location.
- We want to **predict** the **catch score**, as a function of a few covariates:
  - the `Latitude` and `Longitude` of the sampling position. The longitude can be seen as a proxy of the distance from the coast in this specific experiment;
  - the `Depth` of the sea on the sampling position;
  - the `Zone` of the sampling region, either open or closed to **commercial fishing**;
  - the `Year` of the sampling, which can be either `1992` or `1993`.
- Having remove a few observations due to missingness, we split the data into **training** (119 obs.) and **test** set (30 obs.). The full `trawl` dataset is available in the `sm` R package.

The `trawl` dataset

Getting started: linear models

- Let begin our analysis by trying to predict the `Score` using a **linear model** of the form

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}, \qquad i = 1, \ldots, n,$$

- The above values correspond to the variables of the `trawl` dataset, so that

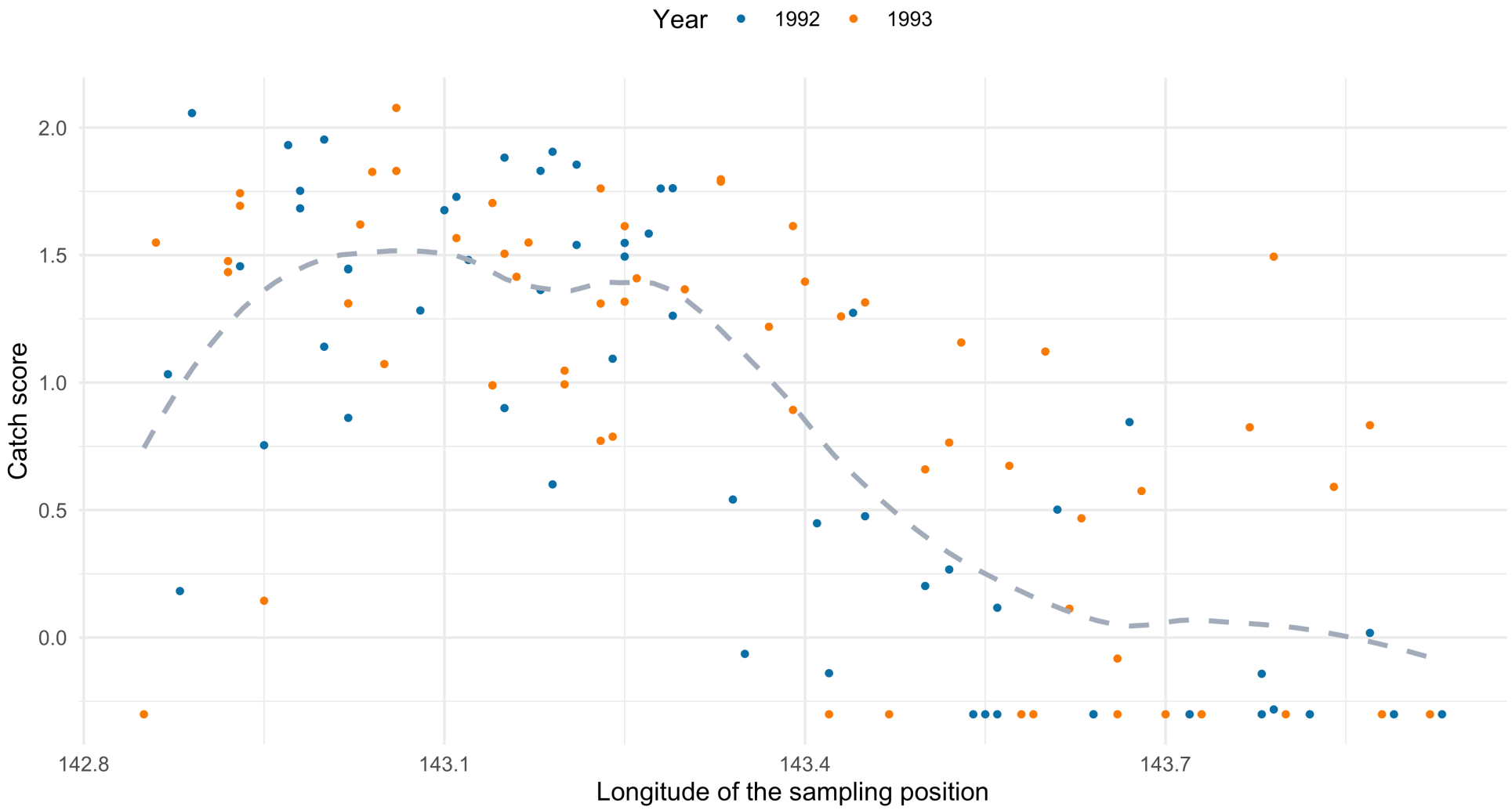
$$\begin{aligned}\texttt{Score}_i = \beta_0 + \beta_1 \texttt{Latitude}_i + \beta_2 \texttt{Longitude}_i + \\ + \beta_3 \texttt{Depth}_i + \beta_4 I(\texttt{Zone}_i = \texttt{Closed}) + \beta_5 I(\texttt{Year}_i = \texttt{1993}).\end{aligned}$$

- Such a model can be estimated using **ordinary least squares**, resulting in:

term	estimate	std.error	statistic	p.value
(Intercept)	297.690	26.821	11.099	0.000
Latitude	0.256	0.222	1.151	0.252
Longitude	-2.054	0.187	-10.955	0.000
Depth	0.020	0.007	3.003	0.003
Zone_Closed	-0.116	0.102	-1.143	0.255
Year_1993	0.127	0.103	1.242	0.217

Home page

Scatterplot with `loess` estimate



Home page

Comments and cricism of linear models

- Is this a good model?
- Granted that every model is just an approximation of reality, it is undeniable that there are some **problematic** aspects.
- By simple graphical inspection, it seems that the relationship between `Score` and `Longitude` is **non-linear**.
- Also, an **interaction effect** between `Year` and `Longitude` could be present.
- These considerations support the idea that a **nonparametric approach** might be more appropriate.
- However, the number of covariates is $p = 5$ and therefore a fully nonparametric estimation would **not** be **feasible**, because of the curse of dimensionality.
- We need a simplified modelling strategy, that accounts for non-linearities but at the same time is not fully flexible.

Generalized additive models (GAM)

The ANOVA decomposition of a function

- We seek for an estimate of (a suitable transformation of) the **mean function**, namely

$$g^{-1}\{\mathbb{E}(Y_i)\} = f(x_{i1}, \ldots, x_{ip}),$$

  where $g^{-1}(\cdot)$ is the so-called **link function**.

- The **unknown** multivariate function $f(\boldsymbol{x}) = f(x_1, \ldots, x_p) : \mathbb{R}^p \to \mathbb{R}$ is **too complex**. However, the following **decomposition** holds

$$f(\boldsymbol{x}) = \beta_0 + \underbrace{\sum_{j=1}^{p} f_j(x_j)}_{\text{Main effect}} + \underbrace{\sum_{j=1}^{p} \sum_{k<j} f_{jk}(x_j, x_k)}_{\text{Interaction effect}} + \underbrace{\sum_{j=1}^{p} \sum_{k<j} \sum_{h<k<j} f_{jkh}(x_j, x_k, x_h)}_{\text{Higher order interaction}} + \cdots .$$

- By imposing **suitable constraints**, this decomposition can be made **unique**.
- More importantly, this decomposition gives us an intuition on how to build non-linear models with a simplified structure.

Generalized additive models (GAM)

- A **generalized additive model** (GAM) presumes a representation of the following type:

$$f(\boldsymbol{x}_i) = \beta_0 + f_1(x_{i1}) + \cdots + f_p(x_{ip}) = \beta_0 + \sum_{j=1}^{p} f_j(x_{ij}), \qquad i = 1, \ldots, n,$$

  where $f_1, \ldots, f_p$ are **smooth univariate** functions with a potentially non-linear behavior.

- In GAMs we include only the **main effects** and we **exclude** the **interactions** terms.
- Generalized linear models (GLMs) are a **special case** of GAMs, in which $f_j(x_{ij}) = \beta_j x_{ij}$.

- To avoid what is essentially a problem of **model identifiability**, it is necessary for the various f_j to be centered around 0, that is

$$\sum_{i=1}^{n} f_j(x_{ij}) = 0, \qquad j = 1, \ldots, p.$$

The backfitting algorithm I

- There exist several strategies for **estimating** the **unknown functions** $f_1, \ldots, f_p$. One of them, called **backfitting**, is particularly appealing because of its elegance and generality.
- Suppose we model each $f_j(x) = \sum_{m=1}^{M_j} \beta_{mj} h_{mj}(x)$ with a **basis expansion**, for example using **regression splines**.
- In a **regression problem** we need to minimize, over the unknown β parameters, the loss

$$\sum_{i=1}^{n} \left\{ y_i - \beta_0 - \sum_{j=1}^{p} f_j(x_{ij}) \right\}^2$$

  subject to the constraint $\sum_{i=1}^{n} f_j(x_{ij}) = 0$.

- When f_j are regression splines, the above loss can be **minimized** using **least squares**. The identifiability issue could be handled by removing the intercept term from each spline basis.
- However, here we consider an **alternative** and **iterative** minimization method, which is similar to the coordinate descent algorithm we employed for the elastic-net.

The backfitting algorithm II

- Now, let us re-arrange the term in the squared loss as follows:

$$\sum_{i=1}^{n} \left\{ y_i - \beta_0 - \sum_{k \neq j} f_k(x_{ik}) - f_j(x_{ij}) \right\}^2 ,$$

  where the highlighted terms are sometimes called **partial residuals**.

- Hence, we can repeatedly and iteratively fit a **univariate smoothing** model for f_j using the **partial residuals** as **response**, keeping fixed the value of the other functions f_k, for $k \neq j$.

- This algorithm produces the same fit of least squares when f_j are regression splines, but the idea is appealing because it can be used with any **generic smoothers** $\mathcal{S}_j$.

- Finally, note that under the constraint $\sum_{i=1}^{n} f_j(x_{ij}) = 0$ the least square estimate for the **intercept** term is $\hat{\beta}_0 = \bar{y}$, i.e. the arithmetic mean.

The backfitting algorithm (regression)

The backfitting algorithm for additive regression models

1. Initialize $\hat{\beta}_0 = \bar{y}$ and set $f_j(x_j) = 0$, for $j = 1, \ldots, p$.
2. Cycle $j = 1, \ldots, p$, $j = 1, \ldots, p$, $\ldots$, until **convergence**:
   i. Update the kth function by smoothing via $\mathcal{S}_j$ the **partial residuals**, so that

$$\hat{f}_j(x) \leftarrow \mathcal{S}_j \left[\left\{ x_{ij}, y_i - \hat{\beta}_0 - \sum_{k \neq j} \hat{f}_k(x_{ik}) \right\}_{i=1}^{n} \right].$$

   ii. Center the function by subtracting its mean

$$\hat{f}_j(x) \leftarrow \hat{f}_j(x) - \frac{1}{n} \sum_{i=1}^{n} \hat{f}_j(x_{ij}).$$

Backfitting: comments and considerations

- The backfitting algorithm, when f_j are modeled as **regression splines**, is known as "Gauss-Seidel". The **convergence** is **guaranteed** under standard conditions.
- Interestingly, even when $\mathcal{S}_j$ are **smoothing splines** the **convergence** of backfitting is **guaranteed**; the proof for this statement is less straightforward.
- In general, however, there is no theoretical guarantee that the algorithm will ever converge, even though the practical experience suggest that this is **not** a **big concern**.
- When $\mathcal{S}_j$ is a **linear smoother** with smoothing matrix $\boldsymbol{S}_j$, then by analogy with the previous unit we can define the **effective degrees of freedom** of $\hat{f}_j$ as

$$\mathrm{df}_j = \mathrm{tr}(\boldsymbol{S}_j).$$

  The number of degrees of the whole model therefore is $\mathrm{df} = 1 + \sum_{j=1}^{p} \mathrm{df}_j$.
- A variant of backfitting for classification problems is available. Once again, relying on **quadratic approximations** of the log-likelihood allows for a generalization to GLMs.

The backfitting algorithm (classification)

Local scoring algorithm for additive logistic regression

1. Initialize $\hat{\beta}_0 = \text{logit}(\bar{y})$ and set $f_j(x_j) = 0$, for $j = 1, \ldots, p$.
2. Iterate **until convergence**:
   i. Define the quantities $\hat{\eta}_i = \hat{\beta}_0 + \sum_{j=1}^{p} \hat{f}_j(x_{ij})$ and $\hat{\pi}_i = \{1 + \exp(-\hat{\eta}_i)\}^{-1}$.
   ii. Construct the **working response**

$$z_i = \hat{\eta}_i + \frac{y_i - \hat{\pi}_i}{\hat{\pi}_i(1 - \hat{\pi}_i)}, \qquad i = 1, \ldots, n.$$

   iii. Construct the **weights** $w_i = \hat{\pi}_i(1 - \hat{\pi}_i)$, for $i = 1, \ldots, n$.
   iv. Use a **weighted backfitting** algorithm using the z_i as responses, which produces a new set of estimates $\hat{f}_1, \ldots, \hat{f}_p$.

GAM using penalized splines

- A common special instance of GAM occurs when **smoothing splines** are employed. In the regression case, the **backfitting** algorithm implicitly minimizes the following **penalized loss**

$$\mathscr{L}(f_1, \ldots, f_p; \lambda) = \sum_{i=1}^{n} \left\{ y_i - \beta_0 - \sum_{j=1}^{p} f_j(x_j) \right\}^2 + \sum_{j=1}^{p} \lambda_j \int_{a_j}^{b_j} \{f_j''(t)\}^2 \mathrm{d}t,$$

  where $\lambda = (\lambda_1, \ldots, \lambda_p)$ is a vector of **smoothing parameters**.

- Each $f_j(x; \beta)$ is a **natural cubic spline**, therefore the penalized least squares criterion is

$$\mathscr{L}(\beta; \lambda) = \sum_{i=1}^{n} \left\{ y_i - \beta_0 - \sum_{j=1}^{p} f_j(x_j; \beta_j) \right\}^2 + \sum_{j=1}^{p} \lambda_j \beta_j^T \mathbf{\Omega}_j \beta_j,$$

  whose joint **minimization** over β is available in closed form.

- Hence, a **direct algorithm** that minimizes $\mathscr{L}(\beta; \lambda)$ is used instead of backfitting.

On the choice of smoothing parameters

- In GAMs there are p **smoothing parameters** $\lambda_1, \ldots, \lambda_p$ that must be selected. We can proceed in the usual way, e.g. considering the **generalized cross-validation** criteria:

$$\mathrm{GCV}(\lambda_1, \ldots, \lambda_p) = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{y_i - \hat{y}_i}{1 - \mathrm{df}/n} \right)^2 .$$

- An alternative criterion in this context is the **REML** (Restricted Maximum Likelihood), which is the **marginal likelihood** of the corresponding **Bayesian model**.

- It is **not possible** to construct a **grid** of values for all the combinations of smoothing parameters $\lambda_1, \ldots, \lambda_p$, because the number of terms increases exponentially in p.

- Hence, many software packages **numerically optimize** the $\mathrm{GCV}(\lambda_1, \ldots, \lambda_p)$, or other information criteria, as a function of $\lambda_1, \ldots, \lambda_p$, using e.g. the Newton-Raphson method.
- Such an approach is particularly convenient in combination with **smoothing splines**, because the **derivatives** needed for Newton's method are available in closed form.

GAM and variable selection

- When p is large there is need to remove the potentially **irrelevant variables**. There exist several **variable selection** ideas for GAMs, but we will not cover the details here.
- **Option 1**. **Stepwise regression**. Perhaps the simplest method, although it is not as efficient as in linear models because we cannot exploit the same computational tricks.
- **Option 2**. **COSSO: Component Selection and Smoothing Operator** (Lin and Zhang, 2006). It's an idea based on combining lasso-type penalties and GAMs.
- **Option 3**. **SpAM: Sparse Additive Models** (Ravikumar, Liu, Lafferty and Wasserman, 2009). Similar to the above, but it exploits a variation of the non-negative garrote.
- **Option 4**. **Double-penalty and shrinkage** (Marra and Wood, 2011). It acts on the penalty term of smoothing splines so that high-values of $\lambda_1, \ldots, \lambda_p$ leads to constant functions.
- **Option X**. **Fancy name**. Yet another method for variable selection with GAMs.

GAM modeling of `trawl` data

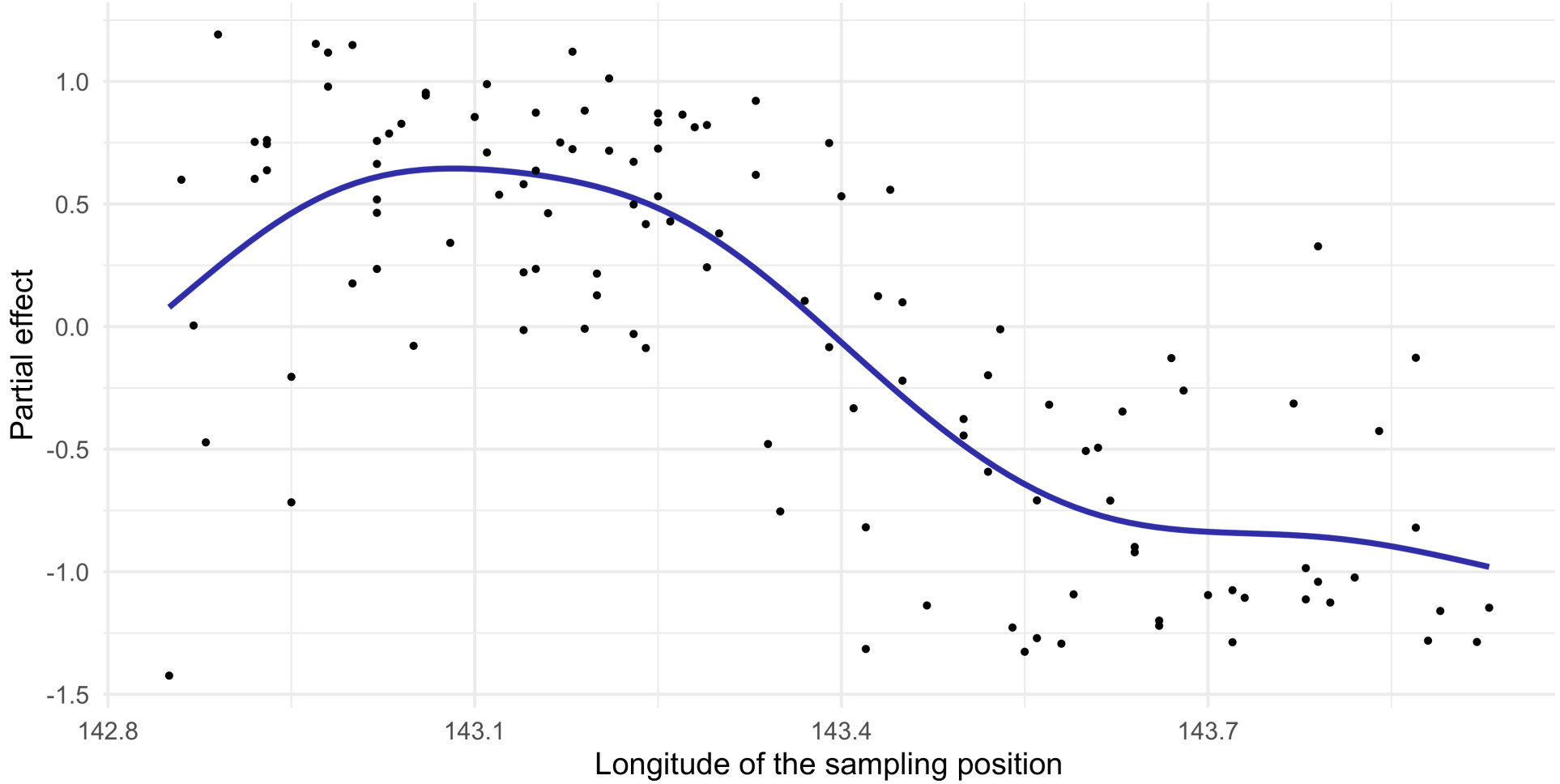
- Let us get back to the `trawl` data. A **specification** based on GAM could be

$$\begin{aligned}\texttt{Score}_i = \beta_0 + f_1(\texttt{Longitude}_i) + f_2(\texttt{Latitude}_i) + f_3(\texttt{Depth}_i) + \\ + \beta_1 I(\texttt{Zone}_i = \texttt{Closed}) + \beta_2 I(\texttt{Year}_i = \texttt{1993}).\end{aligned}$$

- In GAMs the predictors are **not necessarily** modeled using **nonparametric** methods. Indeed, it is common to have a combination of smooth functions and linear terms.
- Besides, it does **not make sense** to "smooth" a **dummy variable**.

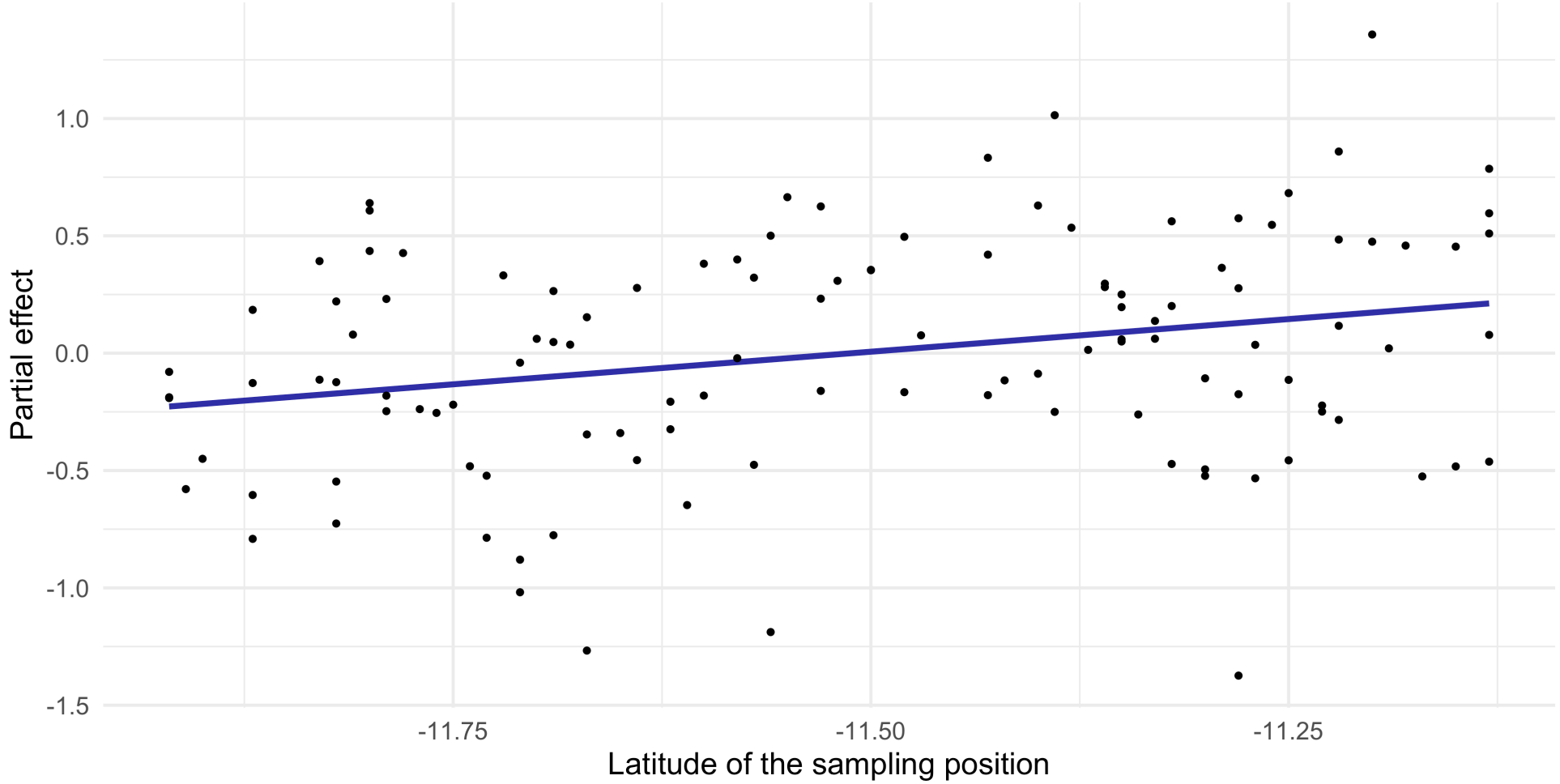
term	estimate	std.error	df
(Intercept)	0.849	0.088	1
Zone_Closed	-0.075	0.099	1
Year_1993	0.149	0.093	1
s(Longitude)	-	-	4.694
s(Latitude)	-	-	1
s(Depth)	-	-	2.447

Partial effect of GAMs (Longitude)



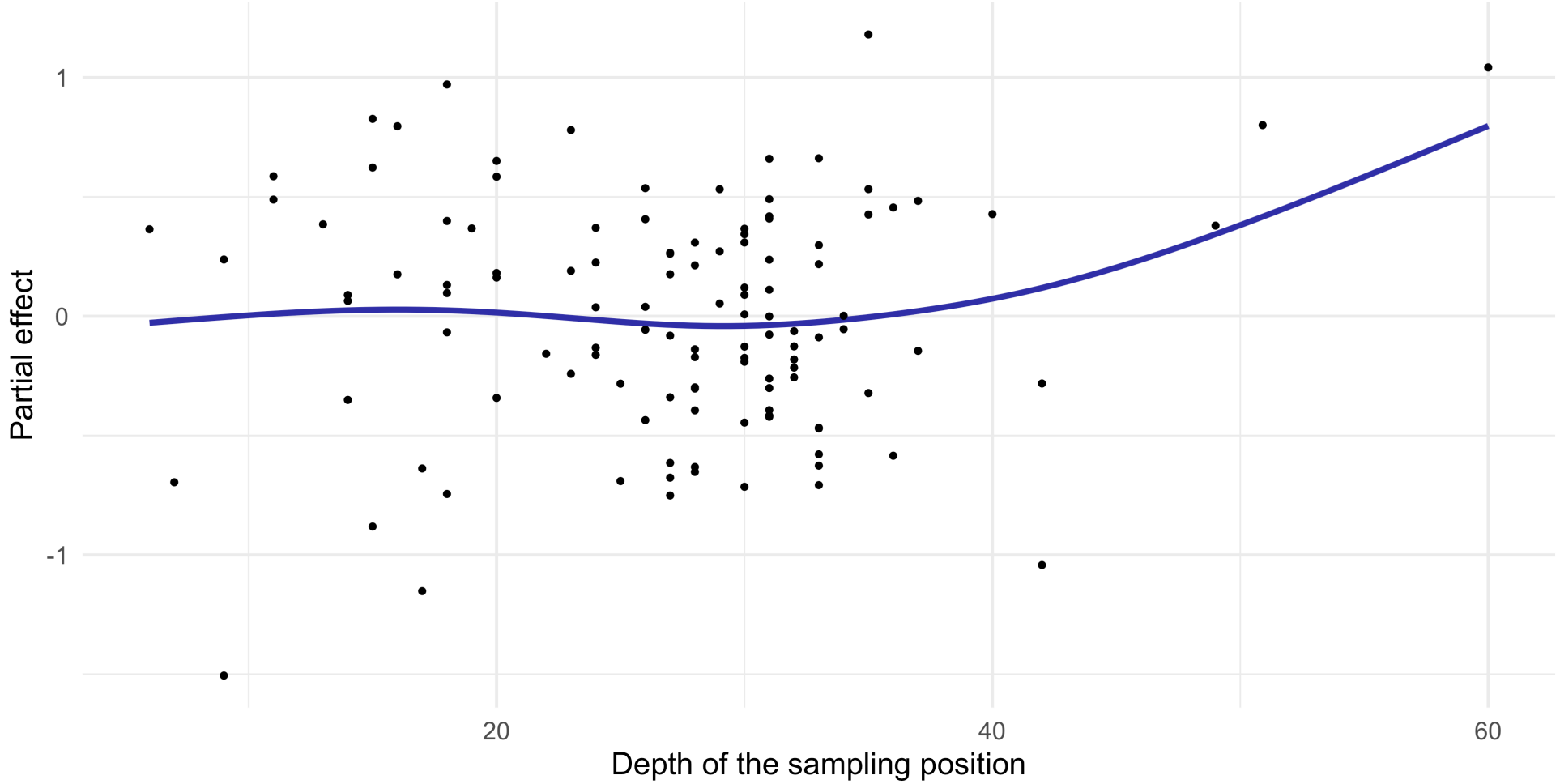
Home page

Partial effect of GAMs (Latitude)



Home page

Partial effect of GAMs (Depth)



Comments and criticism (`trawl` data)

- The fitted GAM model highlights some interesting aspects of the `trawl` data.
- In the first place, it seems confirmed that the `Longitude` has a **marked non-linear** impact on the **catch score**, as the initial analysis was suggesting.
- In particular, the catch score is high when the sampling location is close to the coast (but not too close!), and then it suddenly decreases.
- The effective degrees of freedom of `Latitude` is $\mathrm{df}_2 = 1$, meaning that the estimated $\hat{f}_2$ **collapsed** to a **linear term**. The corresponding shrinkage parameter λ_2 is very high.
- Overall, the effect due to the `Latitude` looks **small** or **not present** at all.
- The `Depth` seems to have a **relevant effect** on the `Score`, but this is likely due to a few **leverage points** at the right extreme of the `Depth` range.
- Finally, we note that both `Zone` and `Year` seem to have a minor effect.

☠ - Naïve Bayes classifier and GAMs

- The **naïve Bayes classifier** expresses the **binary** classification probability $\text{pr}(y = 1 \mid \boldsymbol{x})$ as

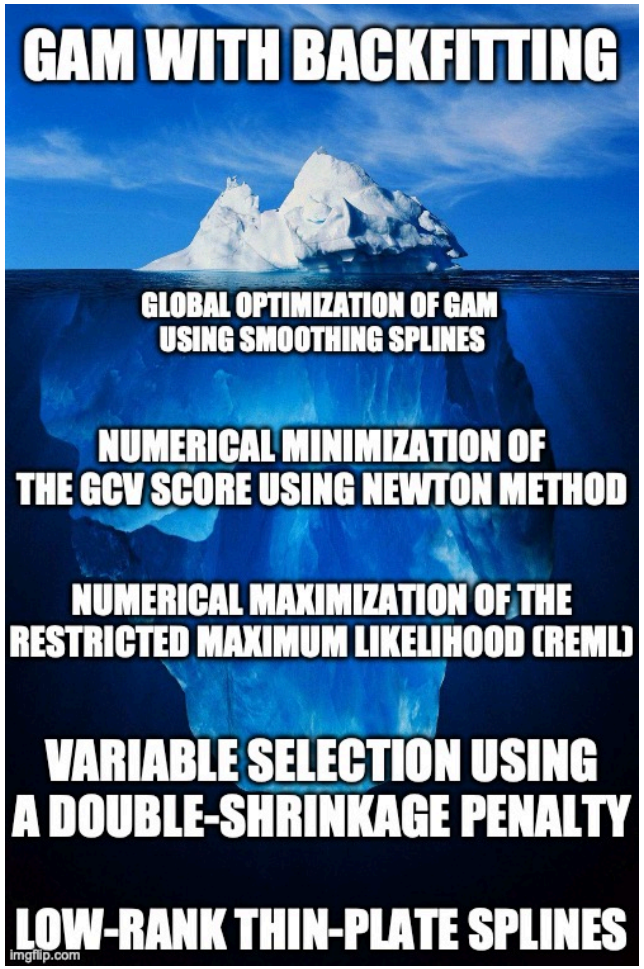
$$\text{pr}(y = 1 \mid \boldsymbol{x}) = \frac{\pi_1 \prod_{j=1}^{p} p_{j1}(x_j)}{\pi_0 \prod_{j=1}^{p} p_{j0}(x_j) + \pi_1 \prod_{j=1}^{p} p_{j1}(x_j)} = \frac{\pi_1 \prod_{j=1}^{p} p_{j1}(x_j)}{p(\boldsymbol{x})}.$$

- Hence, using class 0 as a **baseline**, we can derive the following expression:

$$\log \frac{\text{pr}(y = 1 \mid \boldsymbol{x})}{\text{pr}(y = 0 \mid \boldsymbol{x})} = \log \frac{\pi_1 \prod_{j=1}^{p} p_{j1}(x_j)}{\pi_0 \prod_{j=1}^{p} p_{j0}(x_j)} = \log \frac{\pi_1}{\pi_0} + \sum_{j=1}^{p} \log \frac{p_{j1}(x_j)}{p_{j0}(x_j)} = \beta_0 + \sum_{j=1}^{p} f_j(x_j).$$

- Therefore, although naïve Bayes and GAMs are fitted in a quite different way, there is a **tight connection** among the two methods.
- Naïve Bayes has a **generalized additive model structure**. This also suggests that the "**additive assumption**" is linked to the notion of **independence** among the covariates.

☠ - The mgcv R package



- GAMs were **invented** by Hastie and Tibshirani in 1986, including the backfitting algorithm.
- Simon Wood (2003) described **thin-plate regression splines** and their estimation (no backfitting).
- Simon Wood (2004, 2011) invented methods for estimating $\lambda_1, \ldots, \lambda_p$ in an **efficient** and **stable** manner.
- Marra and Wood (2011) discussed many methods for practical **variable selection** for GAMs.
- For further details, there is a **recent and advanced book** by Simon Wood (2017) entitled "*Generalized Additive Models: An Introduction with R*".
- The `mgcv` package in **R** (by Simon Wood) implements everything mentioned here.

Pros and cons of generalized additive models (GAMs)

Pros

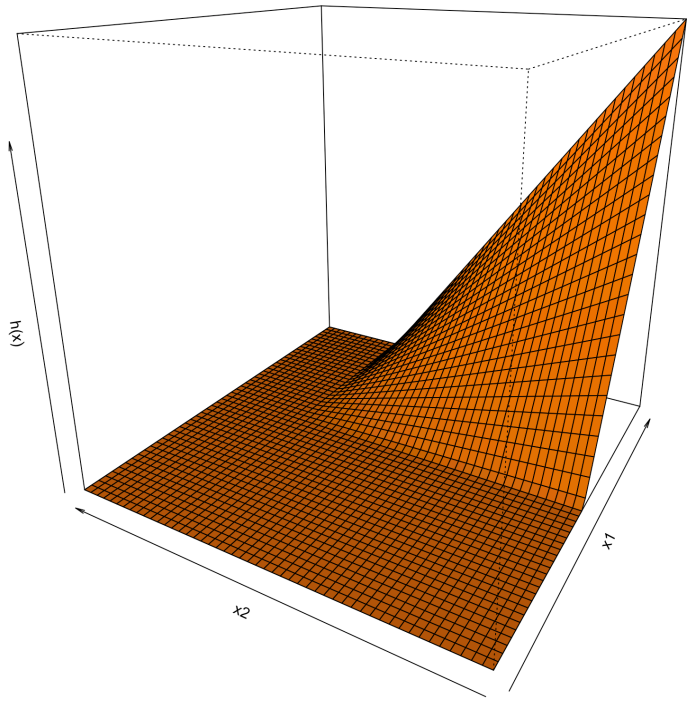
- GAMs can automatically model **non-linear** relationships. This can potentially make more accurate predictions for the response.
- GAMs, as linear models, are **interpretable**: the variation of the fitted response, holding all but one predictor fixed, **does not depend** on the values of the **other predictors**.
- In practice, this means that we can **plot** the **fitted functions** $\hat{f}_j$ separately to examine the roles of the predictors in modelling the response.
- Additive assumption is quite strong, but it is still possible to **manually add interactions** as in the linear regression case.

Cons

- Especially when p is large, it is almost impossible to manually model all the **interactions** among covariates. GAMs do **not** take **second-order** effects (or higher) into account.

MARS

Multivariate Adaptive Regression Splines



- MARS are a generalization of GAMs that avoid the **additivity assumption**.
- MARS allow modeling of **non–linear interactions** and not just non–linear marginal effects.
- MARS are at the same time:
  - A generalization of **stepwise regression**;
  - A method based on multi-dimensional **tensor splines**;
  - A modification of **classification and regression trees** (CART).
- MARS combine many of the techniques we have seen in this course into a single sophisticated algorithm.

MARS additive representation

- MARS is an **additive model** of the form:

$$f(\boldsymbol{x};\beta) = \beta_0 + \sum_{m=1}^{M} \beta_m h_m(\boldsymbol{x}),$$

  where $h_m(\boldsymbol{x})$ are **basis functions** and $\beta = (\beta_1, \ldots, \beta_M)^T$ are regression coefficients.

- Once the basis functions are specified, the estimate for $\hat{\beta}$ is straightforward, using for example **least squares** or the IWLS algorithm in the classification case.
- The main distinction with GAMs is that in MARS the basis functions are **estimated** from the **data** and therefore they are **not pre-specified** in advance.
- MARS is essentially a smart **heuristic algorithm** for selecting a collection of basis functions $h_1(\boldsymbol{x}), \ldots, h_M(\boldsymbol{x})$ that hopefully does not incur in the **curse of dimensionality**.

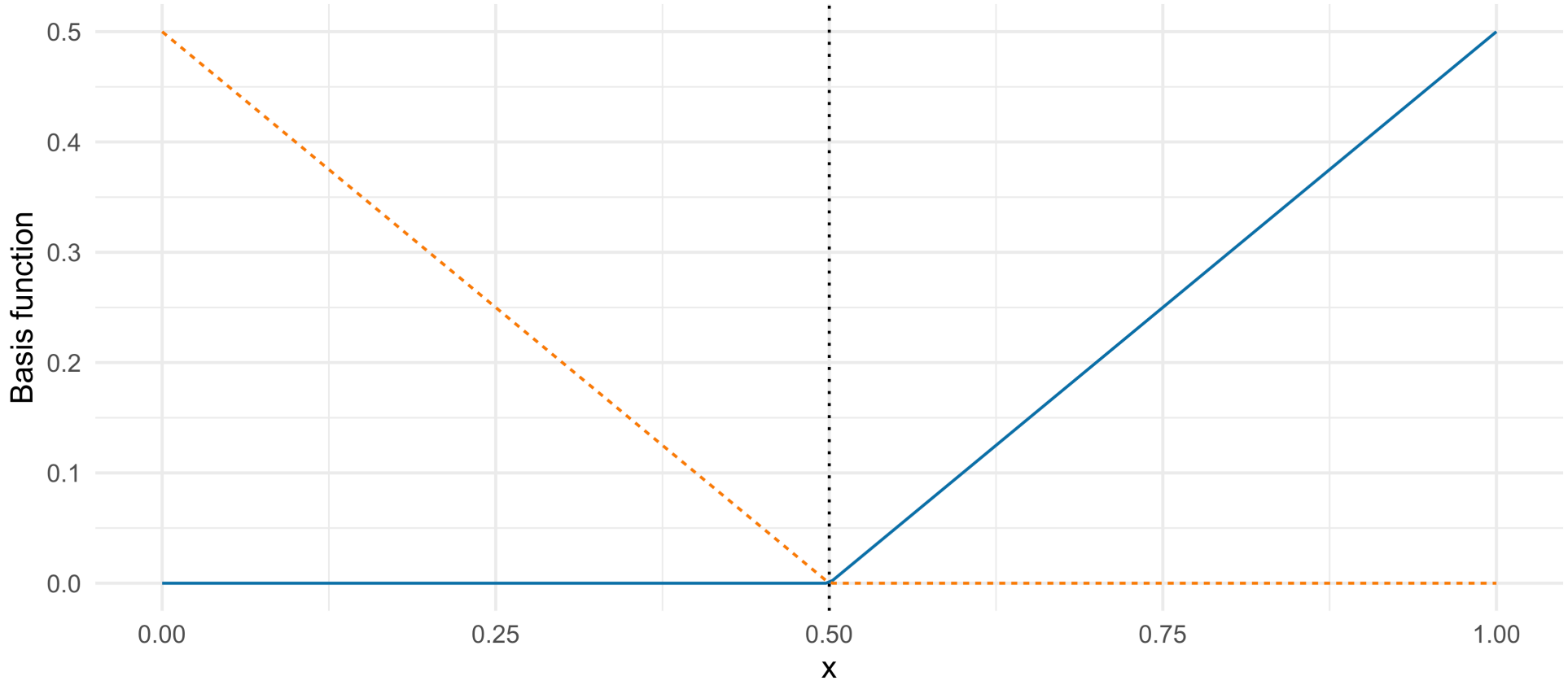
Basis functions for MARS (reflected pairs)

- The MARS algorithm begins by including just the **intercept term**, i.e. $f(\boldsymbol{x}; \beta) = \beta_0$. Then, we proceed by **iteratively** adding basis functions.
- In MARS the basis functions are always **coupled** (or **reflected**), meaning that we always add them in pairs to the additive specification.
- Let us consider the following set of **pairs** of **basis functions** (linear splines):

$$\mathcal{C} = \{(x_j - \xi)_+, (\xi - x_j)_+ : \xi \in \{x_{1j}, \dots, x_{nj}\},\ j = 1, \dots, p\}.$$

  For example, two basis functions could be $h_1(\boldsymbol{x}) = (x_1 - 0.5)_+$ and $h_2(\boldsymbol{x}) = (0.5 - x_1)_+$.
- The **knots** are placed in correspondence of the **observed data**. Hence, there are in **total** $2np$ **possible basis functions** among which we can choose.
- In the **first step** of the MARS algorithm, we identify the pair $h_1(\boldsymbol{x}) = (x_j - \xi)_+$ and $h_2(\boldsymbol{x}) = (\xi - x_j)_+$ that, together with the **intercept**, **minimize** the **loss function**.

An example of reflected pair basis



- The function $h_1(x) = (x - 0.5)_+$ (**blue**) and its reflection $h_2(x) = (0.5 - x)_+$ (**orange**).

A stepwise construction

- Hence, **after** the **first step** of the MARS algorithm, our model for example could be

$$f(\boldsymbol{x};\beta) = \beta_0 + \sum_{m=1}^{2} \beta_m h_m(\boldsymbol{x}) = \beta_0 + \beta_1 (x_1 - 0.5)_+ + \beta_2 (0.5 - x_1)_+.$$

- In the subsequent step, we consider a **new pair** of basis functions $(x_j - \xi)_+, (\xi - x_j)_+$ in $\mathcal{C}$, but this time we are allowed to perform two kind of operations:

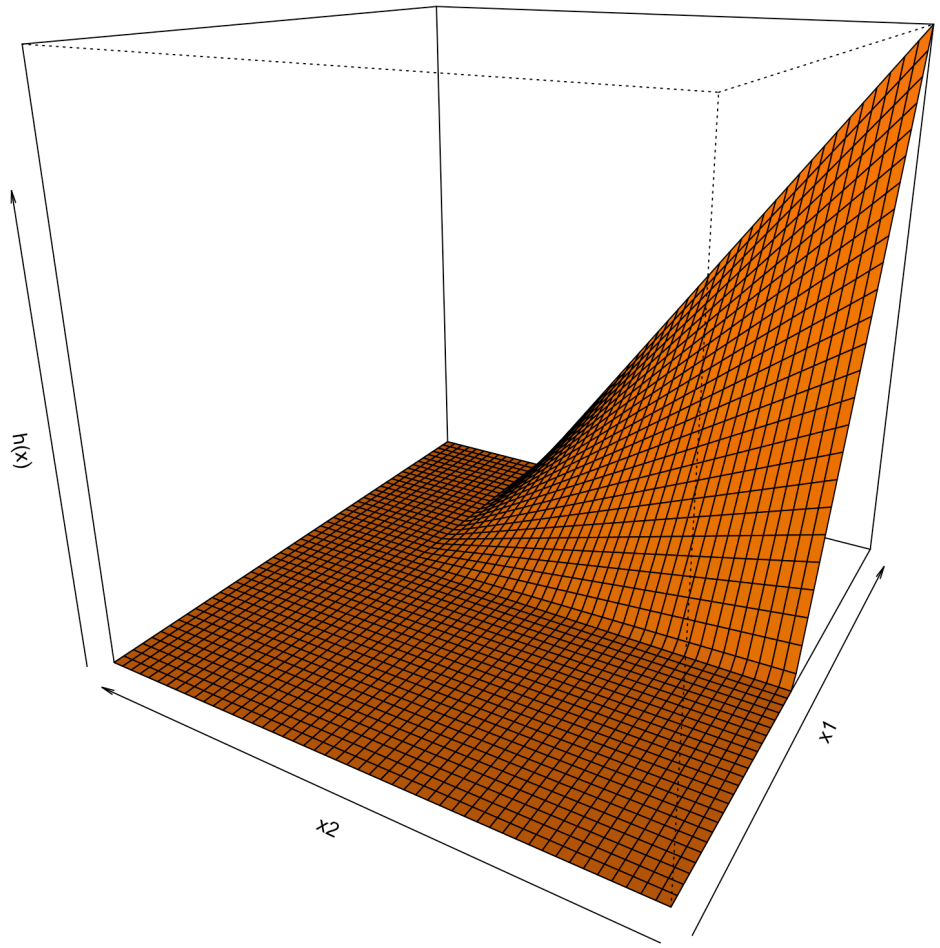
  i. We can include the new pair to the predictor in an **additive** way, obtaining for example

$$f(\boldsymbol{x};\beta) = \beta_0 + \beta_1 (x_1 - 0.5)_+ + \beta_2 (0.5 - x_1)_+ + \beta_3 (x_2 - 0.75)_+ + \beta_4 (0.75 - x_2)_+.$$

  ii. We can include the new pair in a **multiplicative** way, by considering the products between the **new basis** and one of the **old bases** of the model, obtaining for instance

$$\begin{aligned} f(\boldsymbol{x};\beta) = \beta_0 &+ \beta_1 (x_1 - 0.5)_+ + \beta_2 (0.5 - x_1)_+ \\ &+ \beta_3 (x_1 - 0.5)_+ (x_2 - 0.75)_+ + \beta_4 (x_1 - 0.5)_+ (0.75 - x_2)_+. \end{aligned}$$

An example of tensor product basis



- The **product** function $h(\boldsymbol{x}) = (x_1 - 0.5)_+(x_2 - 0.75)_+$ in the range $(0, 1)^2$.

Home page

MARS algorithm (degree d)

1. Initialize $f(\boldsymbol{x}; \beta) = \beta_0$ and let K be **maximum number of pairs**, so that $M = 2K$.
2. Identify the **initial pair** of basis functions h_1 and h_2 in $\mathcal{C}$ that minimize the loss function. Then, let $h_0(\boldsymbol{x}) = 1$ and set $\mathcal{M}_1 = \{h_0, h_1, h_2\}$.
3. For $k = 1, \ldots, K-1$, do:
   i. Let $\mathcal{M}_k = \{h_0, h_1, \ldots, h_{2k}\}$ be the basis functions already present in the model.
   ii. Consider a **novel pair** of bases $\tilde{h}_1, \tilde{h}_2 \in \mathcal{C} \setminus \mathcal{M}_k$. A **candidate pair** is obtained by **multiplying** $\tilde{h}_1, \tilde{h}_2$ with one of the bases in $\mathcal{M}_k$. Note that $h_0 \in \mathcal{M}_k$.
   iii. A **valid** candidate basis does not contain the same variable x_j more than once in the product, and it must involve at most d **product terms**.
   iv. Identify the **optimal pair** among the candidates at steps (ii)-(iii) that **reduces** the **loss function** the most. This results in a new pair of bases h_{2k+1}, h_{2k+2}.
   v. Set $\mathcal{M}_{k+1} \leftarrow \mathcal{M}_k \cup \{h_{2k+1}, h_{2k+2}\}$.
4. Return the collection of models $\mathcal{M}_1, \ldots, \mathcal{M}_K$.

Basis selection and backward regression

- The **degree** d of the MARS algorithms allow to control the number order of interactions of the model. Note that when $d = 1$ it corresponds to a GAM (**no interactions**).
- The final model with M terms is very likely **overfitting** the data. Hence, it is important to remove some of the bases using **backward regression** or best subset.
- The **optimal reduced model** can be selected via cross-validation, but generalized cross-validation is often preferred due to computational reasons.
- Unfortunately, it is not clear how to compute the **effective degrees of freedom** that are needed in the GCV formula.
- In MARS, however, we do not have any miraculous simple formula like in LAR or ridge.
- Simulation studies suggest that, for every knot placed, we should pay a **price** of about 3 **degrees of freedom**. However, this result is quite **heuristic**.

Heuristics behind MARS

- The basis functions used in MARS have the advantage of **operating locally**.
- When the basis functions in $\mathcal{C}$ are multiplied together, the result is **nonzero** only over the small part of the feature space where **both component** functions are **nonzero**.
- Hence, the estimated function is built up **parsimoniously**, by making **small local modifications** to the fit obtained at the previous step.
- This is important, since one should "spend" degrees of freedom carefully in high dimensions, to avoid incurring into the **curse of dimensionality**.
- The constructional logic of the model is **hierarchical**. We can multiply new basis functions that involve new variables only to the basis functions already in the model.
- Hence, an **interaction** of a **higher order** can only be introduced **when** interactions of a **lower order are present**.
- This constraint, introduced for computational reasons, does not necessarily reflect the real behavior of the data, but it often helps in **interpreting the results**.

MARS modeling of `trawl` data ($d = 1$)

- We fit a MARS model with $d = 1$ (**no interactions**), using $M = 20$. Then, the model was simplified using **best subset selection** and GCV. The results are:

Term	Basis function	Coefficient
$h_0(\boldsymbol{x})$	1	1.382
$h_1(\boldsymbol{x})$	$(\texttt{Longitude} - 143.28)_+$	-4.275
$h_2(\boldsymbol{x})$	$(\texttt{Longitude} - 143.58)_+$	3.984

- To clarify, this specification corresponds to the following **estimated** regression **function**:

$$f(\boldsymbol{x}_i; \hat{\beta}) = 1.382 - 4.275(\texttt{Longitude}_i - 143.28)_+ + 3.984(\texttt{Longitude}_i - 143.58)_+,$$

  which has the structure of a GAM. However, the **estimation procedure** is **different**.

- The estimated function $f(\boldsymbol{x}_i; \hat{\beta})$ is **remarkably simple** and it involves only the `Longitude`. Moreover, the relationship between `Score` and `Longitude` is non-linear.
- Both these considerations are consistent with the previous findings, obtained using GAMs.

MARS modeling of `trawl` data ($d = 2$)

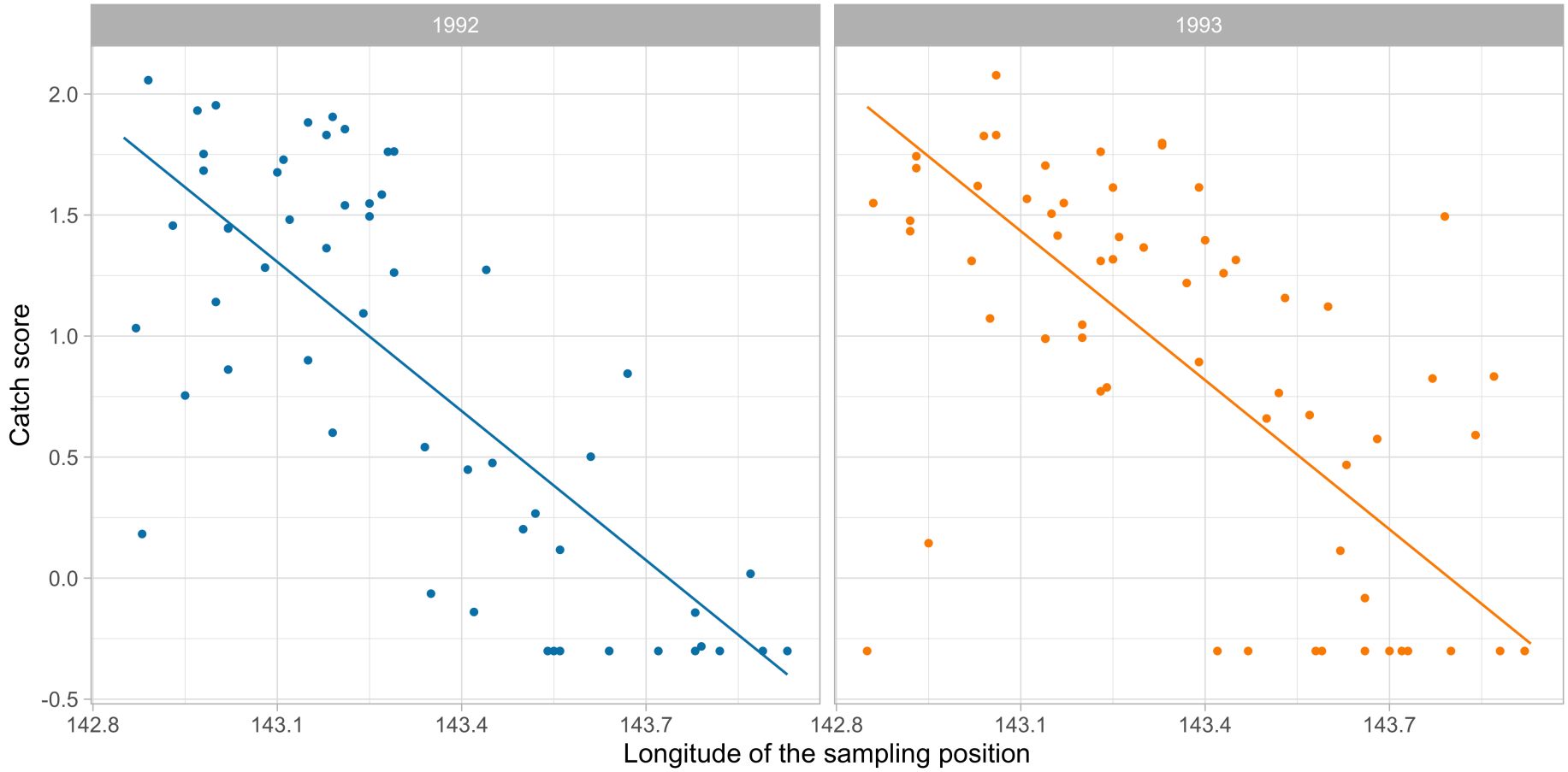
- We fit a MARS model with $d = 2$ (**first order interactions**), using $M = 20$. As before, the model was simplified using **best subset selection** and GCV. The results are:

Term	Basis function	Coefficient
$h_0(\boldsymbol{x})$	1	1.318
$h_1(\boldsymbol{x})$	$(\texttt{Longitude} - 143.28)_+$	-5.388
$h_2(\boldsymbol{x})$	$(\texttt{Longitude} - 143.58)_+$	4.172
$h_3(\boldsymbol{x})$	$I(\texttt{Year} = 1993)(\texttt{Longitude} - 143.05)_+$	0.679
$h_4(\boldsymbol{x})$	$[\texttt{Latitude} - (-11.72)]_+(\texttt{Longitude} - 143.05)_+$	1.489

- As expected, a degree 2 MARS lead to a more sophisticated fit involving **interactions** between `Year` and `Longitude` as well as between `Latitude` and `Longitude`.
- We can explore these effects using **partial plots**.

Partial effects

Linear model | GAM model | MARS (degree 1) | MARS (degree 2)



Home page

Results on the test set

- We can now check the predictive performance on the **test set**, to see which is model scored the best in terms of MAE and RMSE.

	Null model	Linear model	GAM	MARS (degree 1)	MARS (degree 2)
MAE	0.611	0.361	0.315	0.305	0.334
RMSE	0.718	0.463	0.408	0.390	0.407

- It is quite clear that including a **non-linear** specification for the `Longitude` was indeed a good idea that improved the predictive performance.
- Somewhat surprisingly, the **best model** is the extremely simple (yet effective) **MARS of degree 1**.
- The weak interactions effects captured by the MARS of degree 2 led an increased variance of the estimates, slightly deteriorating the fit.

Pros and cons of MARS

Pros

- MARS constructs a sophisticated model by **sequentially refining** the previous fit.
- MARS can capture **interaction** effects.
- The MARS algorithm can be employed for both regression and classification problems.
- The final model is often interpretable. The order of the interactions can be controlled by fixing d.

Cons

- MARS can quickly **overfit** the data if the model complexity is not kept under control.
- Unfortunately, this is not always easy to do as there is **limited theoretical support** that guides us in the choice of the effective degree of freedom.

References

References I

- **Main references**
  - **Chapters 4 and 5** of Azzalini, A. and Scarpa, B. (2011), *Data Analysis and Data Mining*, Oxford University Press.
  - **Chapter 9** of Hastie, T., Tibshirani, R. and Friedman, J. (2009), *The Elements of Statistical Learning*, Second Edition, Springer.
- **Historical references**
  - Hastie, T., and Tibshirani, R. (1986). Generalized Additive Models. *Statistical Science* **1**(3), 209-318.
  - Friedman, Jerome H. 1991. Multivariate Adaptive Regression Splines. *Annals of Statistics* **19**(1), 1-141.

Specialized references I

- **Simon Wood's contributions**
  - Wood, S. N. (2003). Thin plate regression splines. *Journal of the Royal Statistical Society. Series B: Statistical Methodology* **65**, 95-114.
  - Wood, S. N. (2004). Stable and efficient multiple smoothing parameter estimation for generalized additive models. *Journal of the American Statistical Association* **99**, 673-86.
  - Wood, S. N. (2011). Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. *Journal of the Royal Statistical Society. Series B: Statistical Methodology* **73**(1), 3-36.
  - Marra, G. and Wood, S. N. (2011). Practical variable selection for generalized additive models. *Computational Statistics and Data Analysis* **55**(7), 2372–87.
  - Wood, S. N. (2017). *Generalized Additive Models: An Introduction with R*. CRC Press.