Nonparametric Models: Additive Model, Projection Pursuit Regression, CART, MARS

Yanrong Yang

RSFAS/CBE, Australian National University

30th August 2022

Contents of this week

Multivariate Nonparametric Modelling

- Challenge: Curse of Dimensionality
- Solution: Dimension-reduction Modelling
    - Additive Modelling
    - Projection Pursuit Regression
    - Neural Networks
    - Classification and Regression Trees (CART)
    - Multivariate Adaptive Regression Spline (MARS)

Nonparametric Modelling

A multivariate nonparametric model is

$$\mathbb{E}[Y|\mathbf{X} = \mathbf{x}] = f(x_1, x_2, \ldots, x_p). \tag{1}$$

- Traditional Estimation Approaches: Kernel Regression, Basis Expansion (Splines), Local Polynomial Regression, KNN Method.
- As $p > 3$, estimation results from these methods are poor.
- Too much flexibility results in large estimation variance.

Curse of Dimensionality(COD)

As the dimension p is large, nonparametric estimation approaches will fail.

- Parametric Regression: test MSE grows linearly with p;
- Classical Nonparametric Regression: test MSE grows nonlinearly with p, much faster than the parametric case.

Curse of Dimensionality is more severely in nonparametric regression.

Dimension-reduction Models

Recall dimension-reduction models for linear regression.

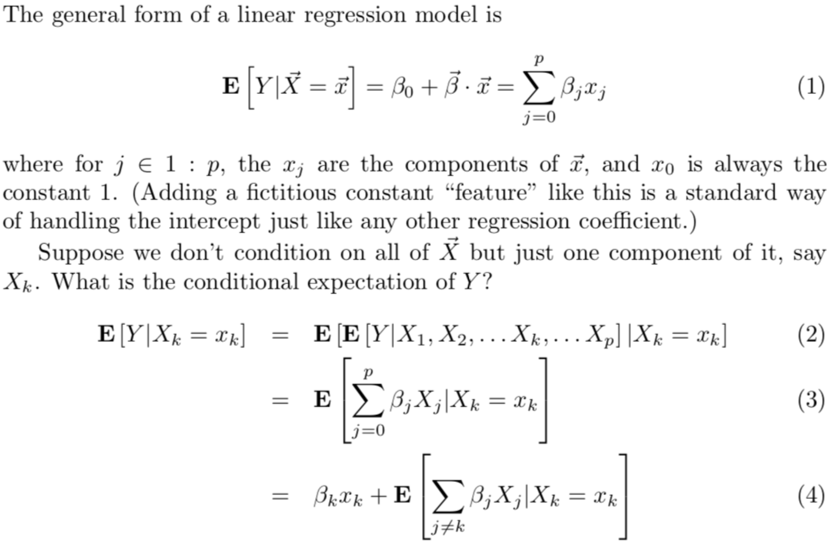
- Principal Component Regression
- Partial Least Squares

We will learn some dimension-reduction models for nonparametric regression.

- Additive Models
- Projection Pursuit Regression
- Neural Networks
- CART and MARS

Additive Models

Motivation from Linear Model



The general form of a linear regression model is

$$\mathbf{E}\left[Y|\vec{X}=\vec{x}\right]=\beta_0+\vec{\beta}\cdot\vec{x}=\sum_{j=0}^{p}\beta_j x_j \tag{1}$$

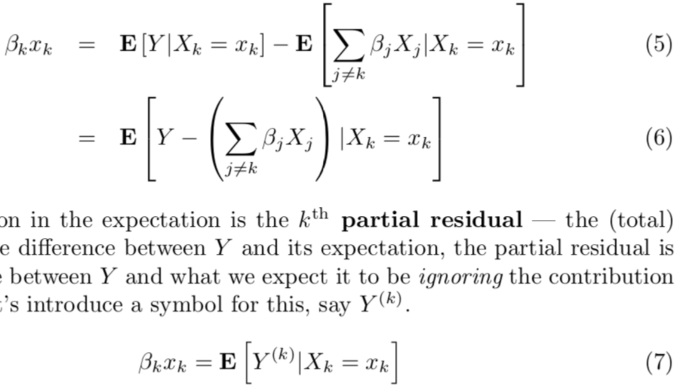
where for $j \in 1:p$, the x_j are the components of $\vec{x}$, and x_0 is always the constant 1. (Adding a fictitious constant “feature” like this is a standard way of handling the intercept just like any other regression coefficient.)

Suppose we don’t condition on all of $\vec{X}$ but just one component of it, say X_k. What is the conditional expectation of Y?

$$\begin{aligned}
\mathbf{E}[Y|X_k=x_k] &= \mathbf{E}\left[\mathbf{E}\left[Y|X_1,X_2,\ldots X_k,\ldots X_p\right]|X_k=x_k\right] && (2)\\
&= \mathbf{E}\left[\sum_{j=0}^{p}\beta_j X_j|X_k=x_k\right] && (3)\\
&= \beta_k x_k+\mathbf{E}\left[\sum_{j\neq k}\beta_j X_j|X_k=x_k\right] && (4)
\end{aligned}$$

Motivation from Linear Model

Partial Residual

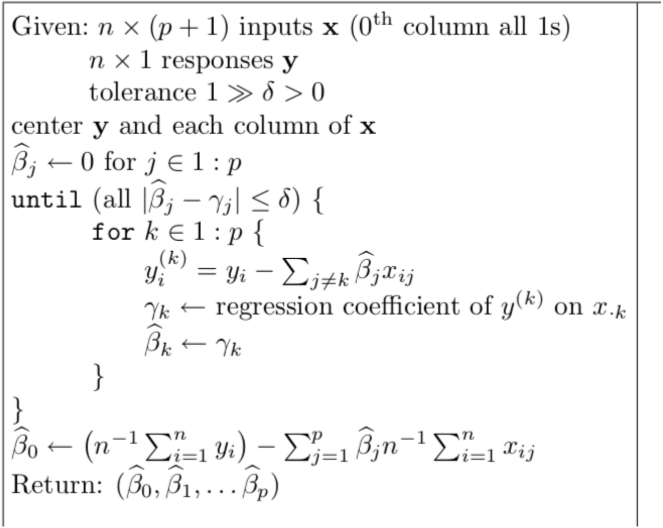


$$\begin{aligned}
\beta_k x_k &= \mathbf{E}\left[Y|X_k = x_k\right] - \mathbf{E}\left[\sum_{j\neq k}\beta_j X_j|X_k = x_k\right] & (5)\\
&= \mathbf{E}\left[Y - \left(\sum_{j\neq k}\beta_j X_j\right)|X_k = x_k\right] & (6)
\end{aligned}$$

The expression in the expectation is the k^{th} **partial residual** — the (total) residual is the difference between Y and its expectation, the partial residual is the difference between Y and what we expect it to be *ignoring* the contribution from X_k. Let's introduce a symbol for this, say $Y^{(k)}$.

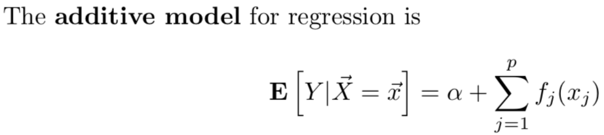
$$\beta_k x_k = \mathbf{E}\left[Y^{(k)}|X_k = x_k\right] \qquad (7)$$

Backfitting Algorithm for Linear Model



```
Given: $n \times (p+1)$ inputs $\mathbf{x}$ ($0^{\text{th}}$ column all 1s)
        $n \times 1$ responses $\mathbf{y}$
        tolerance $1 \gg \delta > 0$
center $\mathbf{y}$ and each column of $\mathbf{x}$
$\widehat{\beta}_j \leftarrow 0$ for $j \in 1:p$
until (all $|\widehat{\beta}_j - \gamma_j| \leq \delta$) {
        for $k \in 1:p$ {
                $y_i^{(k)} = y_i - \sum_{j \neq k} \widehat{\beta}_j x_{ij}$
                $\gamma_k \leftarrow$ regression coefficient of $y^{(k)}$ on $x_{\cdot k}$
                $\widehat{\beta}_k \leftarrow \gamma_k$
        }
}
$\widehat{\beta}_0 \leftarrow (n^{-1} \sum_{i=1}^{n} y_i) - \sum_{j=1}^{p} \widehat{\beta}_j n^{-1} \sum_{i=1}^{n} x_{ij}$
Return: $(\widehat{\beta}_0, \widehat{\beta}_1, \ldots \widehat{\beta}_p)$
```

Additive Model

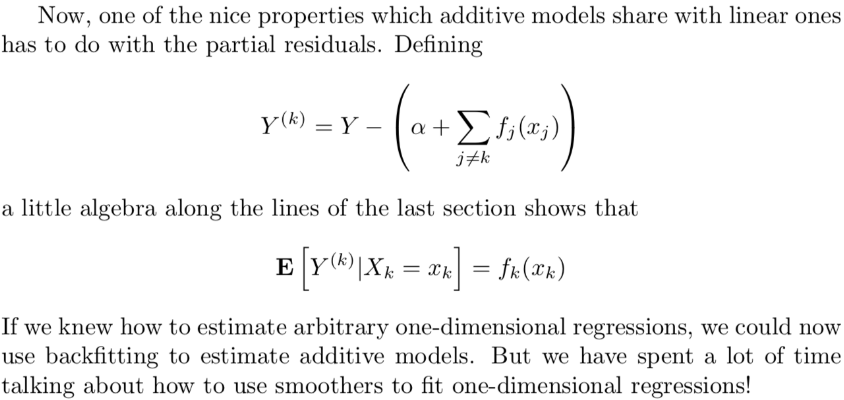


The **additive model** for regression is

$$\mathbf{E}\left[Y|\vec{X}=\vec{x}\right]=\alpha+\sum_{j=1}^{p}f_j(x_j)$$

- Linear Model is a special case: $f_j(x_j)=\beta_j x_j$.
- Identification Conditions: $\mathbb{E}(Y)=\alpha$ and $\mathbb{E}(f_j(X_j))=0$.
- Interpretation: $f_j(\cdot)$ describes how the response variable Y depends on X_j.
- Drawback: no interaction terms among covariates.

Additive Model



Now, one of the nice properties which additive models share with linear ones has to do with the partial residuals. Defining

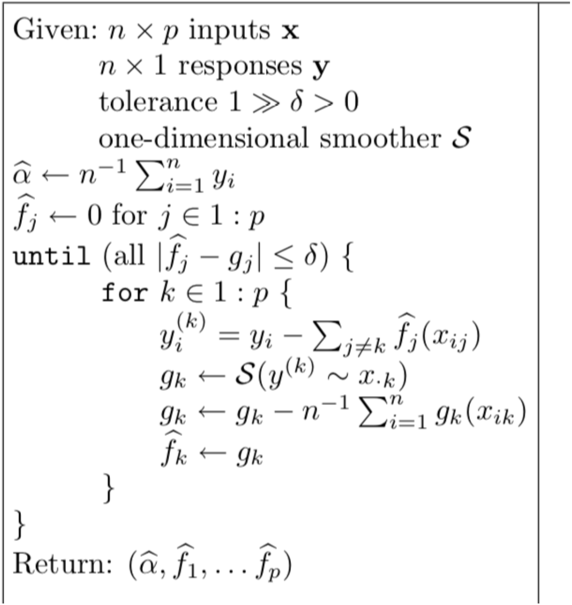
$$Y^{(k)} = Y - \left(\alpha + \sum_{j \neq k} f_j(x_j) \right)$$

a little algebra along the lines of the last section shows that

$$\mathbf{E}\left[Y^{(k)} | X_k = x_k \right] = f_k(x_k)$$

If we knew how to estimate arbitrary one-dimensional regressions, we could now use backfitting to estimate additive models. But we have spent a lot of time talking about how to use smoothers to fit one-dimensional regressions!

Backfitting Algorithm for Additive Model



```
Given: $n \times p$ inputs $\mathbf{x}$
        $n \times 1$ responses $\mathbf{y}$
        tolerance $1 \gg \delta > 0$
        one-dimensional smoother $\mathcal{S}$
$\widehat{\alpha} \leftarrow n^{-1} \sum_{i=1}^{n} y_i$
$\widehat{f}_j \leftarrow 0$ for $j \in 1:p$
until (all $|\widehat{f}_j - g_j| \leq \delta$) {
        for $k \in 1:p$ {
                $y_i^{(k)} = y_i - \sum_{j \neq k} \widehat{f}_j(x_{ij})$
                $g_k \leftarrow \mathcal{S}(y^{(k)} \sim x_{\cdot k})$
                $g_k \leftarrow g_k - n^{-1} \sum_{i=1}^{n} g_k(x_{ik})$
                $\widehat{f}_k \leftarrow g_k$
        }
}
Return: $(\widehat{\alpha}, \widehat{f}_1, \ldots \widehat{f}_p)$
```

Generalized Additive Model (GAM)

- In general, the conditional mean $\mu(X)$ of a response Y is related to an additive function of the predictors via a link function g:

$$g[\mu(X)] = \alpha + f_1(X_1) + \ldots + f_p(X_p)$$

- Examples of classical link functions are the following:
  - $g(\mu) = \mu$ is the identity link, used for linear and additive models for Gaussian response data.
  - $g(\mu) = \text{logit}(\mu)$ as above, or $g(\mu) = \text{probit}(\mu)$, the probit link function, for modeling binomial probabilities. The probit function is the inverse Gaussian cumulative distribution function: $\text{probit}(\mu) = \Phi^{-1}(\mu)$.
  - $g(\mu) = \log(\mu)$ for log-linear or log-additive models for Poisson count data.

Projection Pursuit Regression

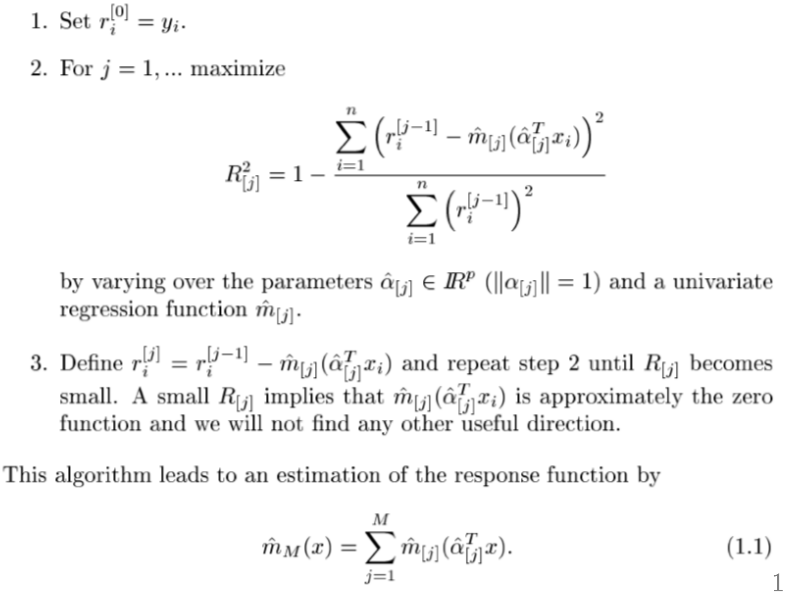
PPR Model

Projection Pursuit Regression Model is

$$\mathbb{E}[Y|X = x] = \sum_{j=1}^{M} m_j(\alpha_j^\top x). \tag{2}$$

- M unknown functions $m_j(\cdot), j = 1, 2, \dots, M$.
- M unknown linear combination vectors $\alpha_j, j = 1, \dots, M$.

PPR Estimation



1. Set $r_i^{[0]} = y_i$.
2. For $j = 1, \ldots$ maximize

$$R^2_{[j]} = 1 - \frac{\sum_{i=1}^{n} \left(r_i^{[j-1]} - \hat{m}_{[j]}(\hat{\alpha}^T_{[j]} x_i) \right)^2}{\sum_{i=1}^{n} \left(r_i^{[j-1]} \right)^2}$$

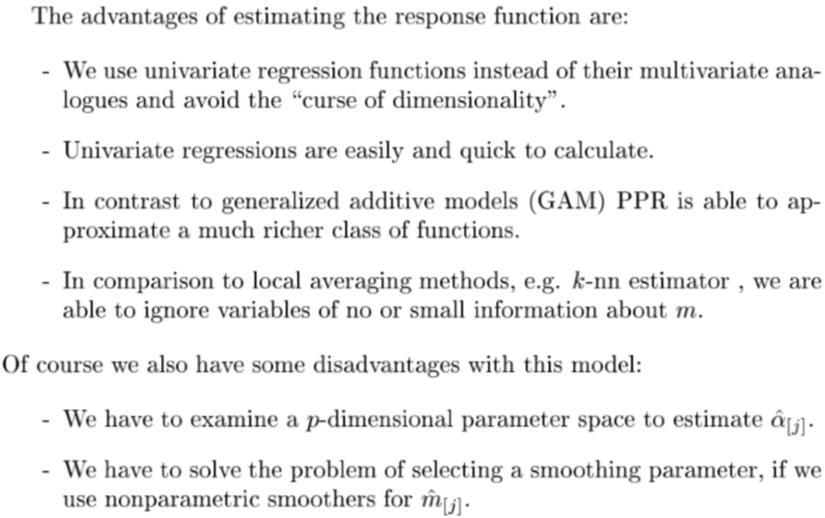
   by varying over the parameters $\hat{\alpha}_{[j]} \in I\!R^p$ $(\|\alpha_{[j]}\| = 1)$ and a univariate regression function $\hat{m}_{[j]}$.

3. Define $r_i^{[j]} = r_i^{[j-1]} - \hat{m}_{[j]}(\hat{\alpha}^T_{[j]} x_i)$ and repeat step 2 until $R_{[j]}$ becomes small. A small $R_{[j]}$ implies that $\hat{m}_{[j]}(\hat{\alpha}^T_{[j]} x_i)$ is approximately the zero function and we will not find any other useful direction.

This algorithm leads to an estimation of the response function by

$$\hat{m}_M(x) = \sum_{j=1}^{M} \hat{m}_{[j]}(\hat{\alpha}^T_{[j]} x). \tag{1.1}$$

Benefits and Drawbacks of PPR



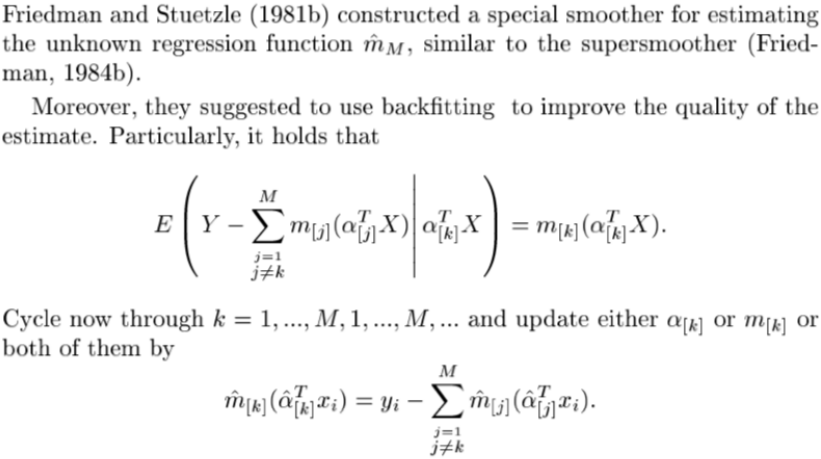
The advantages of estimating the response function are:

- We use univariate regression functions instead of their multivariate analogues and avoid the "curse of dimensionality".
- Univariate regressions are easily and quick to calculate.
- In contrast to generalized additive models (GAM) PPR is able to approximate a much richer class of functions.
- In comparison to local averaging methods, e.g. k-nn estimator , we are able to ignore variables of no or small information about m.

Of course we also have some disadvantages with this model:

- We have to examine a p-dimensional parameter space to estimate $\hat{\alpha}_{[j]}$.
- We have to solve the problem of selecting a smoothing parameter, if we use nonparametric smoothers for $\hat{m}_{[j]}$.

Backfitting on PPR



Friedman and Stuetzle (1981b) constructed a special smoother for estimating the unknown regression function $\hat{m}_M$, similar to the supersmoother (Friedman, 1984b).

Moreover, they suggested to use backfitting to improve the quality of the estimate. Particularly, it holds that

$$E\left(Y - \sum_{\substack{j=1\\ j\neq k}}^{M} m_{[j]}(\alpha_{[j]}^T X) \,\middle|\, \alpha_{[k]}^T X\right) = m_{[k]}(\alpha_{[k]}^T X).$$

Cycle now through $k = 1, ..., M, 1, ..., M, ...$ and update either $\alpha_{[k]}$ or $m_{[k]}$ or both of them by

$$\hat{m}_{[k]}(\hat{\alpha}_{[k]}^T x_i) = y_i - \sum_{\substack{j=1\\ j\neq k}}^{M} \hat{m}_{[j]}(\hat{\alpha}_{[j]}^T x_i).$$

Figure 1a. $Y = X_1X_2 + \epsilon$, $\epsilon \sim N(0, .04)$, *vs.* X_2 *(Y is plotted on the vertical axis,* X_2 *on the horizontal axis. The + symbols represent data points, numbers indicate more than one data point. The smooth is represented by * symbols)*

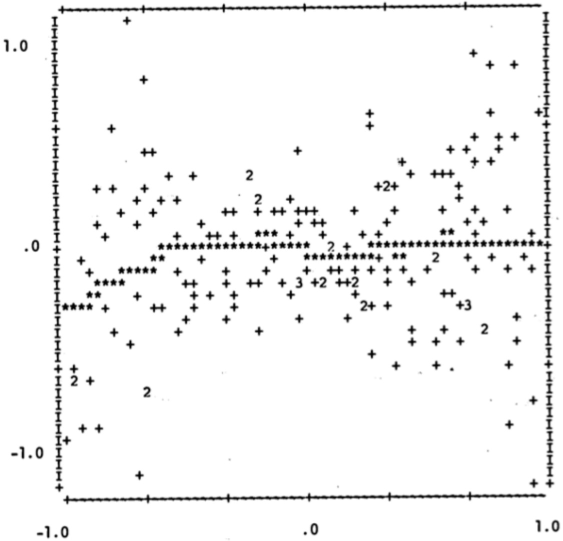


Figure 1b. Y vs. First Solution Linear Combination $\alpha_1 \cdot \boldsymbol{X}, \alpha_1 = (.71, .70)$

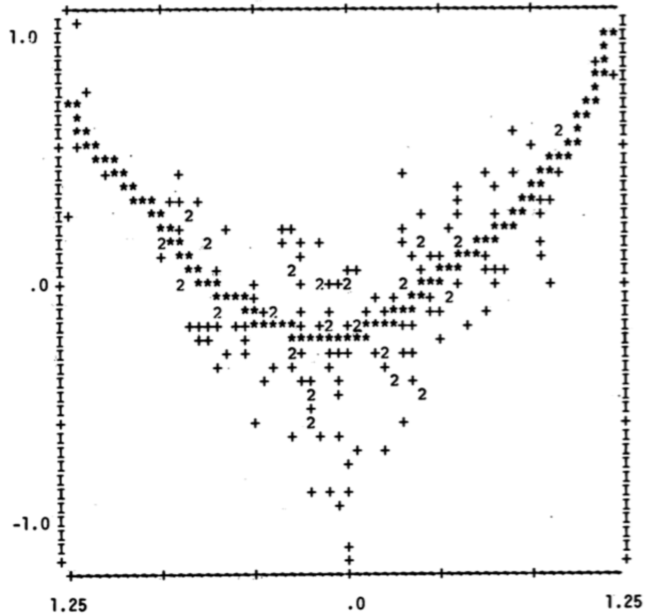


Figure 1c. Residuals From First Solution Smooth vs. Second Solution Linear Combination $\alpha_2 \cdot X$, $\alpha_2 = (.72, -.69)$

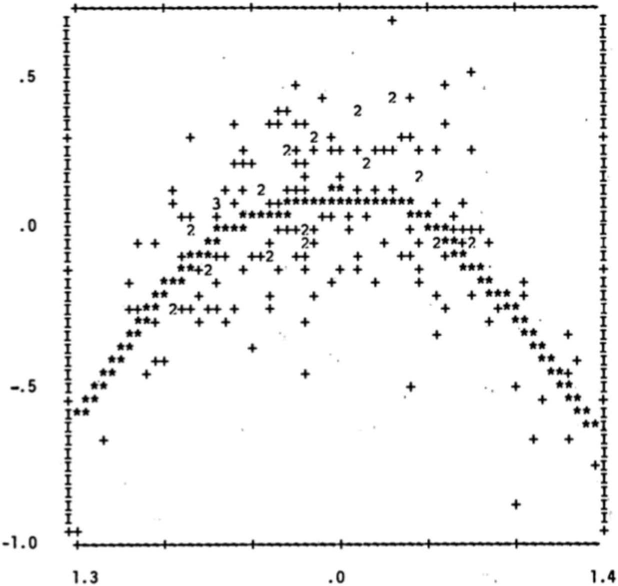
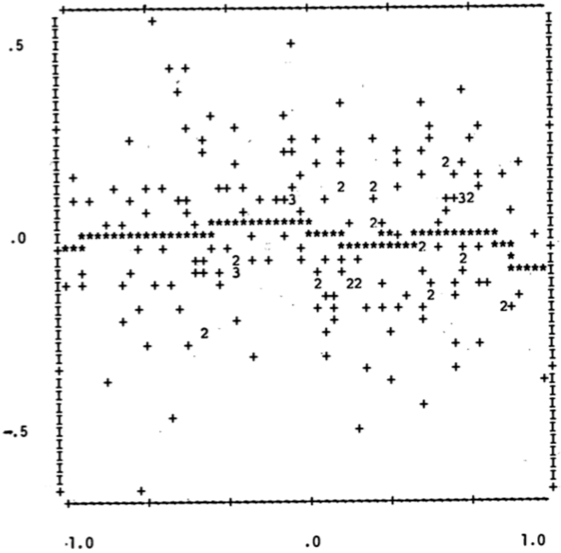
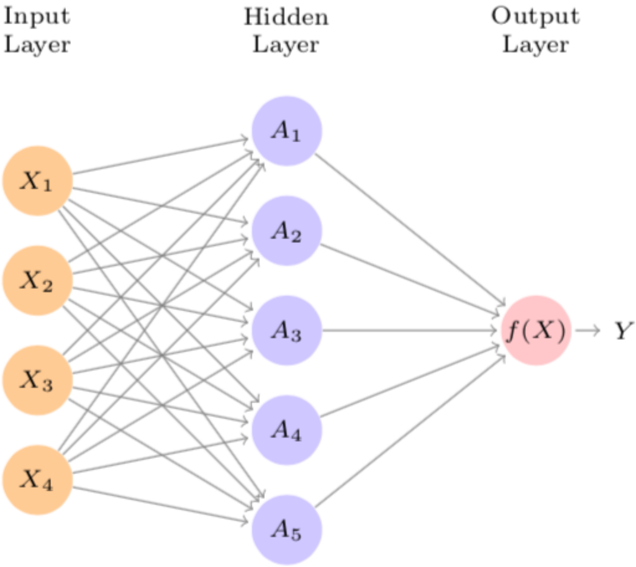


Figure 1d. Residuals From First Two Solution Smooths vs. Third Solution Linear Combination $\boldsymbol{\alpha}_3 \cdot \boldsymbol{X}$, $\boldsymbol{\alpha}_3 = (-.016, .99)$



Neural Networks

Neural Network Structure

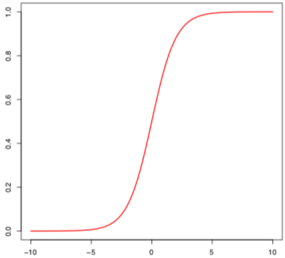


2-Layer Neural Network

Neural networks fit a model of the form

$$Y = \beta_0 + \sum_{j=1}^{r} \gamma_j \psi(\boldsymbol{\beta}_j' \boldsymbol{x} + \nu_j)$$

where ψ is a sigmoidal (or logistic) function and the other parameters (except r) are estimated from the data.



Activation Function

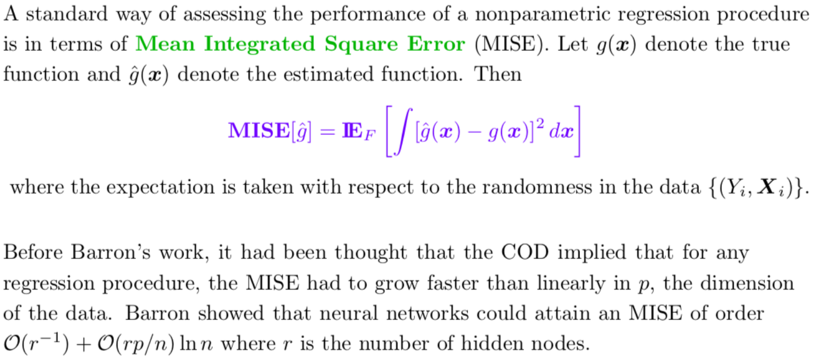
The only difference between PPR and the neural net is that neural nets assumes that the additive functions have a parametric (logistic) form:

$$\psi(\boldsymbol{x}) = \frac{1}{1 + \exp(\alpha_0 + \boldsymbol{\beta}'\boldsymbol{x})}.$$

The parametric assumption allows neural nets to be trained by backpropagation, an iterative fitting technique. This is very similar to backfitting, but somewhat faster because it does not require smoothing.

Barron (1993; *IEEE Transactions on Information Theory*, **39**, 930-945) showed that neural networks evade the Curse of Dimensionality in specific, rather technical, sense. We sketch his result.

Benefit from Neural Network Model and Estimation



A standard way of assessing the performance of a nonparametric regression procedure is in terms of **Mean Integrated Square Error** (MISE). Let $g(\boldsymbol{x})$ denote the true function and $\hat{g}(\boldsymbol{x})$ denote the estimated function. Then

$$\mathbf{MISE}[\hat{g}] = \mathbb{E}_F\left[\int [\hat{g}(\boldsymbol{x}) - g(\boldsymbol{x})]^2 \, d\boldsymbol{x}\right]$$

where the expectation is taken with respect to the randomness in the data $\{(Y_i, \boldsymbol{X}_i)\}$.

Before Barron's work, it had been thought that the COD implied that for any regression procedure, the MISE had to grow faster than linearly in p, the dimension of the data. Barron showed that neural networks could attain an MISE of order $\mathcal{O}(r^{-1}) + \mathcal{O}(rp/n) \ln n$ where r is the number of hidden nodes.

Classification and Regression Trees (CART)

CART Modelling

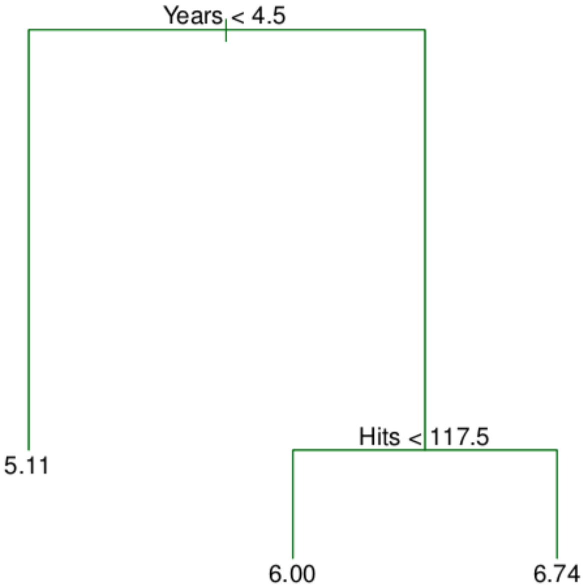
In regression, CART acts as a smart bin-smoother that performs automatic variable selection. Formally, it fits the model

$$Y = \sum_{j=1}^{r} \beta_j I(\boldsymbol{x} \in R_j) + \epsilon$$

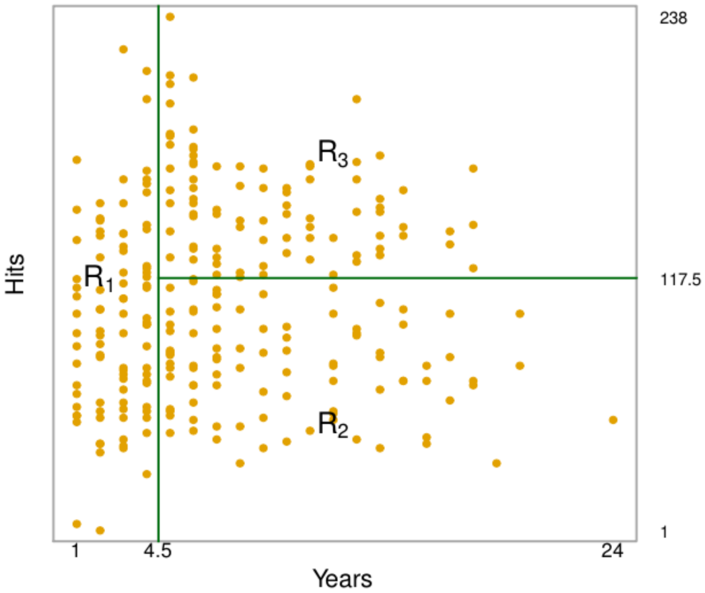
where the regions R_j and the coefficients β_j are estimated from the data. Usually the R_j are disjoint and the β_j is the average of the Y values in R_j.

The CART model produces a decision tree that is helpful in interpreting the results, and this is one of the keys to its enduring popularity.

Example: Regression Tree



Example: Partition Regions



Estimation of CART

The CART algorithm has three parts:

1. A way to select a split at each intermediate node.
2. A rule for declaring a node to be terminal.
3. A rule for estimating Y at a terminal node.

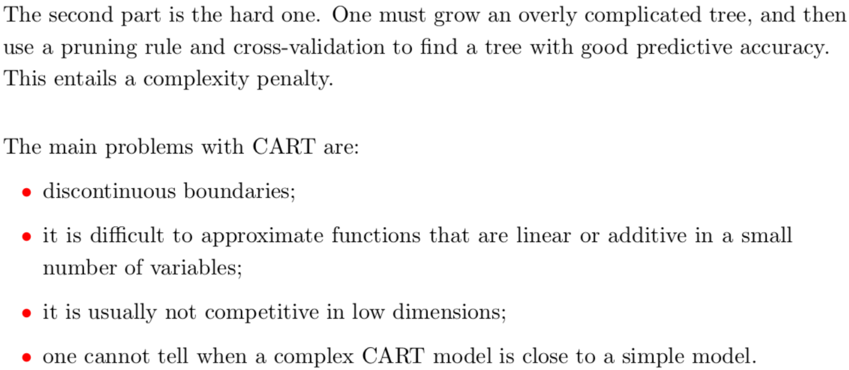
The third part is easy—just use the sample average of the cases at that terminal node.

The first part is also easy—split on the value x_j^* which most reduces

$$SS_{\mathbf{error}} = \sum_{i=1}^{n} (y_i - \hat{f}_c(\boldsymbol{x}_i))^2$$

Where $\hat{f}_c$ is the predicted value from the current tree.

Disadvantage of CART



The second part is the hard one. One must grow an overly complicated tree, and then use a pruning rule and cross-validation to find a tree with good predictive accuracy. This entails a complexity penalty.

The main problems with CART are:

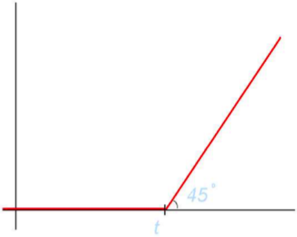
- discontinuous boundaries;
- it is difficult to approximate functions that are linear or additive in a small number of variables;
- it is usually not competitive in low dimensions;
- one cannot tell when a complex CART model is close to a simple model.

Multivariate Adaptive Regression Splines (MARS)

MARS

Multivariate Adaptive Regression Splines (MARS) improves on CART by marrying it to PPR. It uses multivariate splines to let the data find flexible partitions or $\mathbb{R}^p$. And it incorporates PPR by letting the orientation of the region be non-parallel to the natural axes.

The basic building block is a "hockeystick" (first-order truncated basis) function $(x - t)^+$, which looks like:



MARS: Basis Expansion Model

The fitted model has the form

$$Y = \sum_{j=1}^{r} \beta_j B_m(\boldsymbol{x}) + \epsilon$$

where

$$B_m(\boldsymbol{x}) = \prod_{k \in \mathcal{K}} [s_{km}(x_{km} - t_{km})]^+$$

for $s_k m = \pm 1$ and $\mathcal{K}$ is a subset of the explanatory variables. Thus B_m is a product of hockeysticks, so $\hat{f}$ is continuous. The regions are determined by the knots $\{t_{km}\}$.

The MARS algorithms starts with $B_1(\boldsymbol{x}) = 1$ and constructs new terms until there are too many, as measured by generalized cross-validation.

Empirically, Friedman found that each term fit in a MARS model costs between 2 and 4 degrees of freedom. This reflects the variable selection involved in the fitting, but not smoothing.

Conclusion

- In nonparametric regression, dimension reduction is more necessary than linear regression.
- Dimension-reduction models include additive model, projection pursuit regression, neural networks, cart and mars.
- Estimation approaches for these methods, except neural networks at this stage.