Chapter 3. Spline smoothing and semi-parametric Models (II)
Part 3

March 24, 2007

1 Some comments in applying gam

package gam can only tell us the estimated components $g_1(.), ..., g_p(.)$ in model

$$Y = \beta_0 + g_1(X_1) + ... + g_p(X_p) + \varepsilon$$

Since $Eg_1(X_1) = 0, ..., Eg_p(X_p) = 0$, we have immediately

$$\beta_0 = EY$$

It can be estimated by

$$\hat{\beta}_0 = n^{-1} \sum_{i=1}^{n} Y_i.$$

The gam can also be applied to simple regression

$$Y = g(X) + \varepsilon$$

In this case, the estimated g by gam is actually $g(.) - E(Y)$.

2 Application of polynomial spline to other models

The idea is to approximate any function by linear combination of polynomials. Thus, we only need to estimate a linear regression model to estimate to original model. Consider the varying coefficient model

$$Y = a_0(Z) + a_1(Z)X_1 + a_2(Z)X_2 + \varepsilon.$$

Suppose $B_1(Z), ..., B_{J+4}(Z)$ is the basis for all functions of variable Z. Then, we have approximately

$$a_0(Z) = \sum_{j=1}^{J+4} \theta_{0,j} B_j(Z) \qquad a_1(Z) = \sum_{j=1}^{J+4} \theta_{1,j} B_j(Z) \qquad a_2(Z) = \sum_{j=1}^{J+4} \theta_{2,j} B_j(Z)$$

and the original model becomes (linearized "linear model")

$$Y = \sum_{j=1}^{J+4} \theta_{0,j} B_j(Z) + \sum_{j=1}^{J+4} \theta_{1,j} \{B_j(Z) X_1\} + \sum_{j=1}^{J+4} \theta_{2,j} \{B_j(Z) X_2\} + \varepsilon.$$

The coefficients are

$$\mathbf{B} = (\theta_{0,1}, ..., \theta_{0,J+4}, \theta_{1,1}, ..., \theta_{1,J+4}, \theta_{2,1}, ..., \theta_{2,J+4})^\top.$$

Suppose we have observations

$$\begin{aligned}
Y_1 &= a_0(Z_1) + a_1(Z_1) X_{11} + a_2(Z_1) X_{12} + \varepsilon_1, \\
Y_2 &= a_0(Z_2) + a_1(Z_2) X_{21} + a_2(Z_1) X_{22} + \varepsilon_2, \\
&... \\
Y_n &= a_0(Z_n) + a_1(Z_n) X_{n1} + a_2(Z_n) X_{n2} + \varepsilon_n
\end{aligned}$$

The corresponding linear model is then

$$\begin{aligned}
Y_1 &= \sum_{j=1}^{J+4} \theta_{0,j} B_j(Z_1) + \sum_{j=1}^{J+4} \theta_{1,j} \{B_j(Z_1) X_{11}\} + \sum_{j=1}^{J+4} \theta_{2,j} \{B_j(Z_1) X_{12}\} + \varepsilon_1 \\
Y_2 &= \sum_{j=1}^{J+4} \theta_{0,j} B_j(Z_2) + \sum_{j=1}^{J+4} \theta_{1,j} \{B_j(Z_2) X_{21}\} + \sum_{j=1}^{J+4} \theta_{2,j} \{B_j(Z_1) X_{12}\} + \varepsilon_2 \\
&... \\
Y_n &= \sum_{j=1}^{J+4} \theta_{0,j} B_j(Z_n) + \sum_{j=1}^{J+4} \theta_{1,j} \{B_j(Z_n) X_{n1}\} + \sum_{j=1}^{J+4} \theta_{2,j} \{B_j(Z_n) X_{n2}\} + \varepsilon_n
\end{aligned}$$

Let

$$\mathbf{X} = \begin{pmatrix}
B_1(Z_1) & B_2(Z_1) & ... & B_{J+4}(Z_1) & B_1(Z_1)X_{11} & ... & B_{J+1}(Z_1)X_{11} & B_1(Z_1)X_{12} & ... & B_{J+1}(Z_1)X_{12} \\
B_1(Z_2) & B_2(Z_2) & ... & B_{J+4}(Z_2) & B_1(Z_2)X_{21} & ... & B_{J+1}(Z_2)X_{21} & B_1(Z_1)X_{22} & ... & B_{J+1}(Z_2)X_{22} \\
... & & & & & & & & & \\
B_1(Z_n) & B_2(Z_n) & ... & B_{J+4}(Z_n) & B_1(Z_n)X_{n1} & ... & B_{J+1}(Z_n)X_{n1} & B_1(Z_n)X_{n2} & ... & B_{J+1}(Z_n)X_{n2}
\end{pmatrix}$$

$$\mathbf{Y} = \begin{pmatrix} Y_1 \\ Y_2 \\ ... \\ Y_n \end{pmatrix}, \quad \mathcal{E} = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ ... \\ \varepsilon_n \end{pmatrix}$$

By LSE, the estimated coefficients are

$$\begin{aligned}\hat{\mathbf{B}} &= (\hat{\theta}_{0,1}, ..., \hat{\theta}_{0,J+4}, \hat{\theta}_{1,1}, ..., \hat{\theta}_{1,J+4}, \hat{\theta}_{2,1}, ..., \hat{\theta}_{2,J+4})^{\rightarrow} \\ &= (\mathbf{X}^\top \mathbf{X})^{-1}(\mathbf{X}^\top \mathbf{Y})\end{aligned}$$

The estimated functions are

$$\hat{a}_0(z) = \sum_{j=1}^{J+4} \hat{\theta}_{0,j} B_j(z) = (B_1(z), ..., B_{J+4}(z), 0, ..., 0, 0, ..., 0)\hat{\mathbf{B}}$$

$$\hat{a}_1(z) = \sum_{j=1}^{J+4} \hat{\theta}_{1,j} B_j(z) = (0, ..., 0, B_1(z), ..., B_{J+4}(z), 0, ..., 0)\hat{\mathbf{B}}$$

$$\hat{a}_2(z) = \sum_{j=1}^{J+4} \hat{\theta}_{2,j} B_j(z) = (0, ..., 0, 0, ..., 0, B_1(z), ..., B_{J+4}(z))\hat{\mathbf{B}}$$

The fitted values are

$$\hat{Y}_i = \hat{a}_0(Z_i) + \hat{a}_1(Z_i)X_{i1} + \hat{a}_2(Z_i)X_{i2}$$

If further $\mathcal{E} \sim N(0, \sigma^2 I)$, then

$$\hat{\mathbf{B}} - \mathbf{B} \sim N(0, (\mathbf{X}^\top \mathbf{X})^{-1}\sigma^2)$$

where σ^2 can be esitmated by

$$\hat{\sigma}^2 = n^{-1} \sum_{i=1}^{n} \{Y_i - \hat{Y}_i\}^2$$

The estimated coefficient fucntions are then

$$\hat{a}_0(z) - a_0(z) \sim N(0, (B_1(z), ..., B_{J+4}(z), 0, ..., 0, 0, ..., 0)(\mathbf{X}^\top \mathbf{X})^{-1}(B_1(z), ..., B_{J+4}(z), 0, ..., 0, 0, ..., 0)^\top \sigma^2)$$

$$\hat{a}_1(z) - a_1(z) \sim N(0, (0, ..., 0, B_1(z), ..., B_{J+4}(z), 0, ..., 0)(\mathbf{X}^\top \mathbf{X})^{-1}0, ..., 0, B_1(z), ..., B_{J+4}(z), 0, ..., 0)^\top \sigma^2)$$

$$\hat{a}_2(z) - a_2(z) \sim N(0, (0, ..., 0, 0, ..., 0, B_1(z), ..., B_{J+4}(z))(\mathbf{X}^\top \mathbf{X})^{-1}(0, ..., 0, 0, ..., 0, B_1(z), ..., B_{J+4}(z))^\top \sigma^2)$$

Consider more complicated models, for example

$$Y = a_0(Z) + a_1(Z)X_1 + a_2(W)X_2 + \varepsilon.$$

3 Multidimensional splines

We have discussed using the polynomial spline to regression with one independent variable. The splines can also be extended to high dimensional regressions. suppose we have independent variables $\mathbf{x}_1, ..., \mathbf{x}_p$. Again we are interested in $m(x_1, ..., x_p) = E(Y|\mathbf{x}_1 = x_1, ..., \mathbf{x}_p = x_p)$. for each variable, we may have cubic polynomial basis

$$\text{Basis for } \mathbf{x}_k : B_{k,1}(x_1), B_{k,2}(x_k), ..., B_{k,J_k+4}(x_1)$$

Denoted by $\mathcal{B}_k$. then the basis for a p dimensional function is

$$\mathcal{B}_1, ..., \mathcal{B}_p, \underbrace{\mathcal{B}_1 \times \mathcal{B}_2, ..., \mathcal{B}_{p-1} \times \mathcal{B}_p}_{\text{interaction terms}}, ...,$$

Denote them by

$$h_{1,k}(x_i), h_{2,k}(x_i, x_j), h_{3,j}(x_i, x_j, x_k)$$

A spline function is of the form

$$s(x_1, ..., x_p) = \sum \alpha_k h_{1,k}(x_i) + \sum \beta_k h_{2,k}(x_i, x_j) + ...$$

Suppose we have sample $(\mathbf{x}_{i1}, ..., \mathbf{x}_{p1}, Y_i), i = 1, ..., n$. To estimate the model, we use Least-square estimation and minimize

$$\sum_{i=1}^{n} \{Y_i - s(X_i)\}^2$$

The problem with this approach is that we have too many terms and the model might be over-fitted. A simple approach to avoid this is by adding a penalty term (leading to ridge regression).

4 Multivariate Adaptive Regression Splines (MARS)

Multivariate Adaptive Regression Splines (MARS) is an implementation of techniques popularized by Friedman (1991) for solving regression-type problems (see also, Multiple Regression), with the main purpose to predict the values of a continuous dependent or outcome variable from a set of independent or predictor variables.

MARS used expansions in piecewise linear basis functions of the form $(x_k - t)_+$ and $(t - x_k)_+$ (called reflected pair). This is actually a first order polynomial basis, where t is the **knot**.

Therefore, the collection of the basis functions are

$$\mathcal{C} = \{(x_k - t)_+, (t - x_k)_+ : t \in \{x_{1k}, ..., x_{Jk}\}, k = 1, ..., p\}$$

If we consider a model of linear combination of $\mathcal{C}$. Then it is actually an approximation for additive model

$$Y = g(\mathbf{x}_1) + ... + g(\mathbf{x}_p) + \varepsilon$$

If we hope to include the interaction term among the independent variables (e.g. in model $Y = \mathbf{x}_1 + \mathbf{x}_2 + \mathbf{x}_1 \sin(\mathbf{x}_2) + \varepsilon$), we need to consider the interaction of basis functions for example

$$(x_k - t_{k,\iota})_+ * (x_j - t_{k,\kappa})_+$$

If we include higher order interaction, we can that there are huge number of basis functions. Denote the basic function by $h_m(x), m = 1, 2, ..., M$

The final model is assumed as

$$Y = \beta_0 + \sum_{m=1}^{M} \beta_m h_m(X) + \varepsilon$$

forward model selection of the basis Suppose we have samples $(X_i, Y_i) = (\mathbf{x}_{i1}, ..., \mathbf{x}_{ip}, Y_i)$, $i = 1, ..., n$. Consider any model of the form

$$Y = \beta_0 + \beta_1 a(X) + ... + \beta_m a_m(X) + \varepsilon \tag{4.1}$$

We need to decide whether an additional term from possible candidates, say $a_{m+1,1}(X), ..., a_{m+1,n_1}(X)$, should be included in the model where $a_k(x)$ are known functions. that is to compare model (4.1) with

$$Y = \beta_0 + \beta_1 a(X) + ... + \beta_m a_m(X) + \beta_{m+1,k} a_{m+1,k}(X) + \varepsilon, \quad k = 1, ..., n_1. \tag{4.2}$$

For this purpose, we need to calculate the CV values for all the $(1 + n_1)$ models.

$$CV_k(m + 1) = n^{-1} \sum_{i=1}^{n} \{Y_i - \hat{f}^{\setminus i}_{m+1,k}(X_i)\}^2$$

where

$$\hat{f}^{\backslash i}_{m+1,k}(X_i) = \hat{\beta}^{\backslash i}_0 + \hat{\beta}^{\backslash i}_1 a(X_i) + ... + \hat{\beta}^{\backslash i}_m a_m(X_i) + \hat{\beta}^{\backslash i}_{m+1,k} a_{m+1,k}(X_i)$$

$\hat{\beta}^{\backslash i}_0, ..., \hat{\beta}^{\backslash i}_{m+1}$ are the Least square estimator of the model (4.2) with observation i deleted. This calculation might be not a easy job!

Let's consider the Least square estimation (without deleting any observations) and let

$$\mathbf{X} = \begin{pmatrix} 1 & a(X_1) & ... & a_{m+1,k}(X_1) \\ 1 & a(X_2) & ... & a_{m+1,k}(X_2) \\ ... & & & \\ 1 & a(X_n) & ... & a_{m+1,k}(X_n) \end{pmatrix}, \quad \mathbf{Y} = \begin{pmatrix} Y_1 \\ Y_2 \\ ... \\ Y_n \end{pmatrix}$$

Then the fitted Y_i is

$$\hat{\mathbf{Y}} = \begin{pmatrix} \hat{f}_{m+1,k}(X_1) \\ \hat{f}_{m+1,k}(X_2) \\ ... \\ \hat{f}_{m+1,k}(X_2) \end{pmatrix} = \mathcal{S}\mathbf{Y}$$

It is proved that

$$Y_i - \hat{f}^{\backslash i}_{m+1,k}(X_i) = \frac{Y_i - \hat{f}_{m+1,k}(X_i)}{1 - s_{ii}}$$

where s_{ii} is the (i,i)th entry of $\mathcal{S}$. Thus

$$\begin{aligned} CV_k(m+1) &= n^{-1} \sum_{i=1}^{n} \Big[\frac{Y_i - \hat{f}_{m+1,k}(X_i)}{1 - s_{ii}} \Big]^2 \\ &\approx n^{-1} \sum_{i=1}^{n} [Y_i - \hat{f}_{m+1,k}(X_i)]^2 / (1 - sum^n_{i=1} s_{ii}/n) \\ &= \sum_{i=1}^{n} [Y_i - \hat{f}_{m+1,k}(X_i)]^2 / (1 - n_p/n)^2 \\ &= \frac{RSS}{(1 - n_p/n)^2} \stackrel{def}{=} GCV_k(m+1) \end{aligned}$$

where n_p is the number of coefficients in the model. If

$$k_0 = \arg_k \min GCV_k(m+1)$$

Let $GCV(m+1) = GCV_{k_0}(m+1)$. If $GCV(m+1) < GCV(m)$, we introduce term $a_{m+1,k_0}(X)$ into the model; otherwise, we stop and the final model is (4.1). Continue the procedure until no more terms can be added to the model

For MARS, we can apply the above idea to select the spline basis functions by starting with model

$$Y = \beta_0 + \varepsilon \tag{4.3}$$

Each time we add one term into the model.

backward pruning

After we selected a term and get a model

$$Y = \beta_0 + \beta_1 a(X) + ... + \beta_m a_m(X) + \beta_{m+1,k_0} a_{m+1,k_0}(X) + \varepsilon \tag{4.4}$$

we need to check whether there is any term that can be eliminated. Again, we can implement this procedure based on GCV.

MARS use a binary splitting to add new basis functions

we start with only the constant function $h_0(x) = 1$ in our model as in (4.3). We consider the candidates in $\mathcal{C}$ and set $\mathcal{M} = \{h_0(x)\}$. At each stage, we consider a new basis function pair all products of function h_m in the model set $\mathcal{M}$ with one of the reflected pairs in $\mathcal{C}$. We add to the model $\mathcal{M}$ the term of the form

$$\hat{\beta} h_\ell(X) * (\mathbf{x}_\iota - t)_+ + \hat{\beta}' h_\ell(X) * (t - \mathbf{x}_\iota)_+, \quad h_\ell \in \mathcal{M}$$

that produces the largest decrease in training error RSS.

The back pruning procedure is also necessary here.

Example 4.1 (Simulation) *100 samples are drawn from*

$$Y = \mathbf{x}_1 * \mathbf{x}_2 + 0.2\varepsilon$$

where $\mathbf{x}_1, \mathbf{x}_2, \varepsilon$ *are IID N(0,1).*

MARS can give a good approximation to the regression surface as in figure 1

Note that the PPR (with 2 components) and MARS models are both correct. However, if we calculate the CV values, MARS seems to have smaller CV value than PPR model.

Example 4.2 (ozone) **(data)** *The level of ozone might be affected by radiation, temperature and wind. consider models*

(1) $ozone^{1/3} = g_1(rad.) + g_2(temp.) + g_3(wind) + \varepsilon$

(2) $MARS$

Their CV values are 0.2380925, 0.2568687 respectively. thus, model (1) is selected. **(code)**

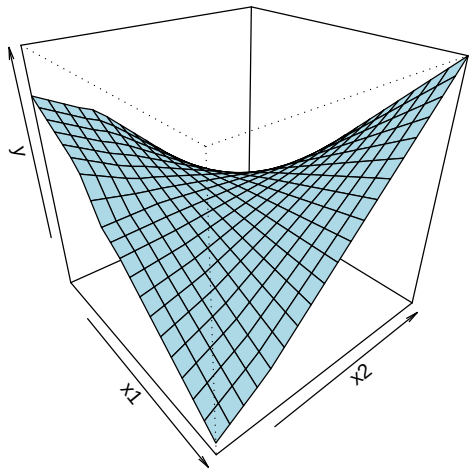

Figure 1: The estimated GAM model **(code)**

Example 4.3 *In the Prostate data, predictors are lcavol, lweight, age, lbph, svi, lcp, gleason, pgg45. The response variable is lpsa. For* **training data** *apply CART, GAM, MARS, LM (linaer regression model), PPR to built a model. Based on these models, predict the lpsa in* **validation data**.

The prediction errors are MARS: 2.500208, LM: 1.887553, CART: 2.74196, and GAM: 1.771182 with model

$$lpsa \sim lcavol + lweight + s(age, 2) + s(lbph, 2) + svi + lcp + s(gleason, 2) + pgg45$$

and PPR(2 components): 2.704825

(code1) (code2) (code3) (code4) (code5)

Based on these values, we can say a GAM model as above is the best amongst all the candidate models.

References

J. Friedman (1991) Multivariate Adaptive Regression Splines (with discussion) *Annals of Statistics*, *19*/1, 1-141.