

Additive Models

Smoothers estimate models of the form

$$Y = f(x) + \varepsilon$$

for a single or low-dimensional x .

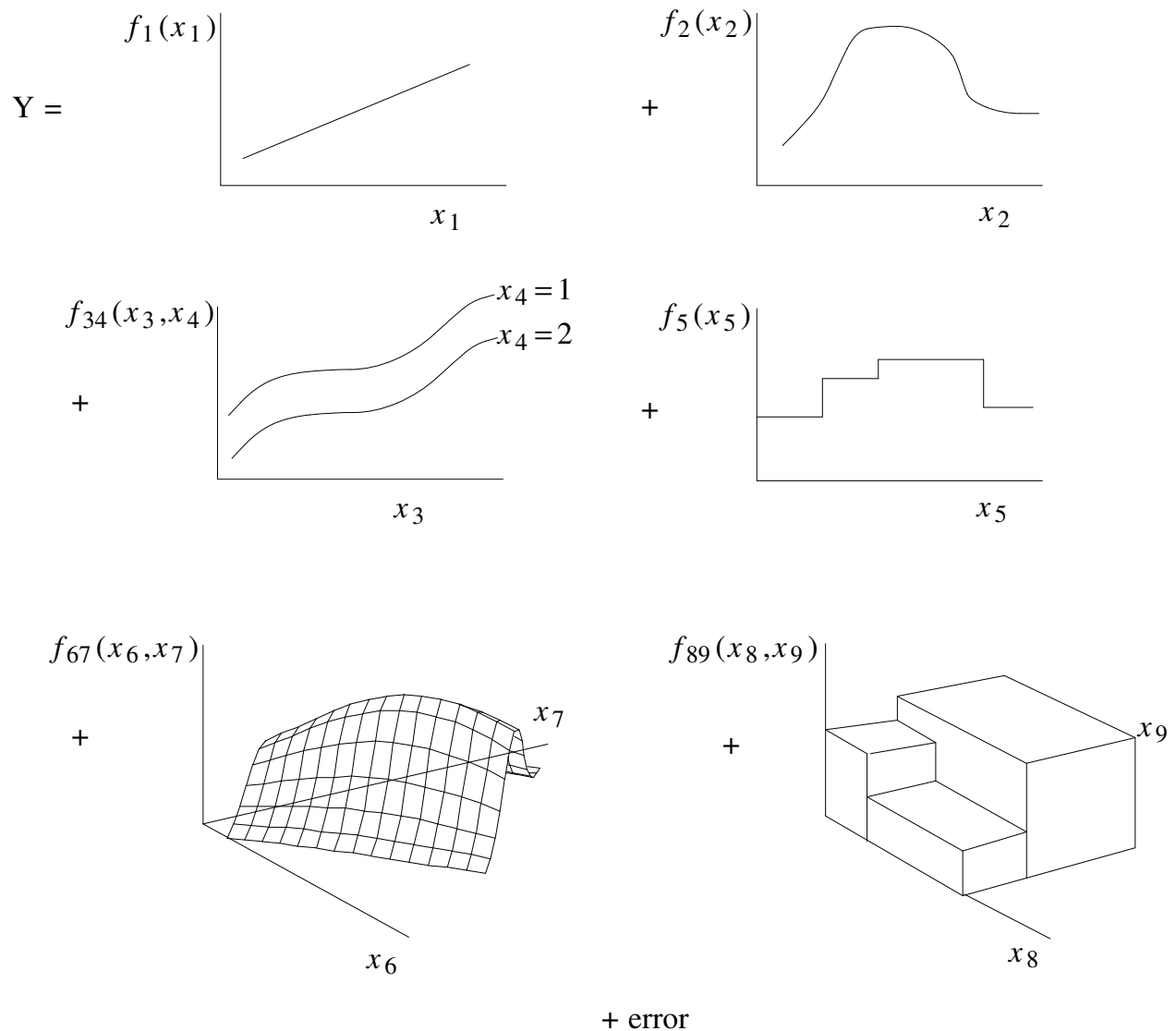
Additive models allow us to use smoothers as building blocks to fit models of the form

$$Y = \alpha + f_1(x_1) + f_2(x_2) + \cdots + f_p(x_p) + \varepsilon$$

Examples

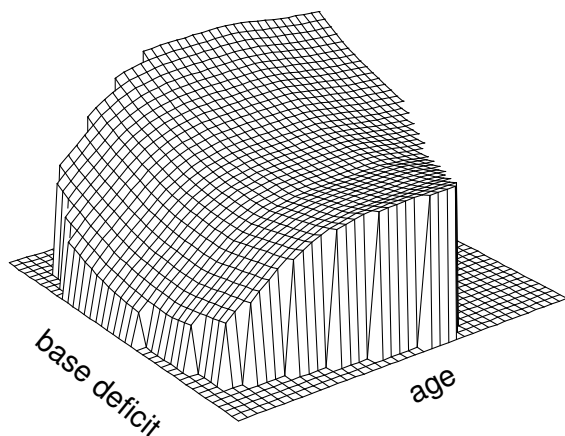
- $Y = \alpha + \beta_1 x_1 + \beta_2 x_2 + \varepsilon$ — linear model
- $Y = \alpha + \{\beta_{11} x_1 + \beta_{12} x_1^2\} + \beta_2 \log(x_2) + \sum_j \beta_{3j} 1_{(c_j < x_3 \leq c_{j+1})} + \varepsilon$
- $Y = \alpha + \beta_1 x_1 + \beta_2 x_2 + f_3(x_3) + \varepsilon$ — semiparametric model
- $Y = \alpha + \beta_1 x_1 + \beta_2 x_2 + f_3(x_3) + f_4(x_4) + \varepsilon$
- $Y = \alpha + \beta_1 x_1 + f_2(x_2) + f_{34}(x_3, x_4) + \varepsilon$

ANOVA Pictures

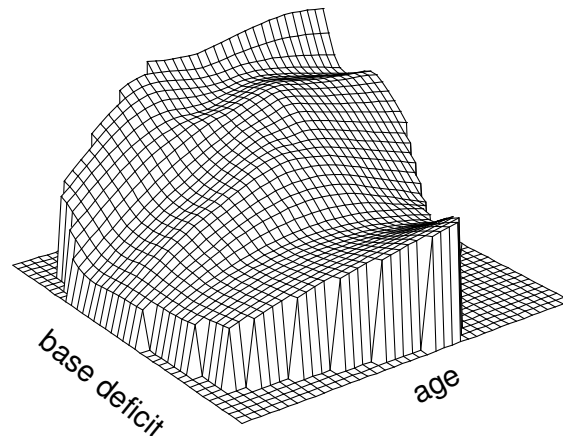


The price for additivity

$$f_1(\text{age}) + f_2(\text{base deficit})$$



$$f(\text{age}, \text{base deficit})$$



Data from a study of diabetic children, predicting log C-peptide (a blood measurement). See GAM book, chapter 2.

Left plot: for each value of **age**, the function of **base deficit** has the same shape — the level differs. And vice versa.

Right plot: a two-dimensional smoother appears to have found an interaction (turns out not to be significant).

Generalized Additive Models

Two-class Logistic Regression

A model for 0/1 data, as in classification problems with 2 classes. $P(Y = 1|X) = \text{Binomial}(1, p(X))$

$$\begin{aligned} \text{logit } p(x) &= \log\left(\frac{p(x)}{1 - p(x)}\right) \\ &= \alpha + f_1(x_1) + f_2(x_2) + \cdots + f_p(x_p) \end{aligned}$$

or

$$p(x) = \frac{e^{\alpha + f_1(x_1) + f_2(x_2) + \cdots + f_p(x_p)}}{1 + e^{\alpha + f_1(x_1) + f_2(x_2) + \cdots + f_p(x_p)}}$$

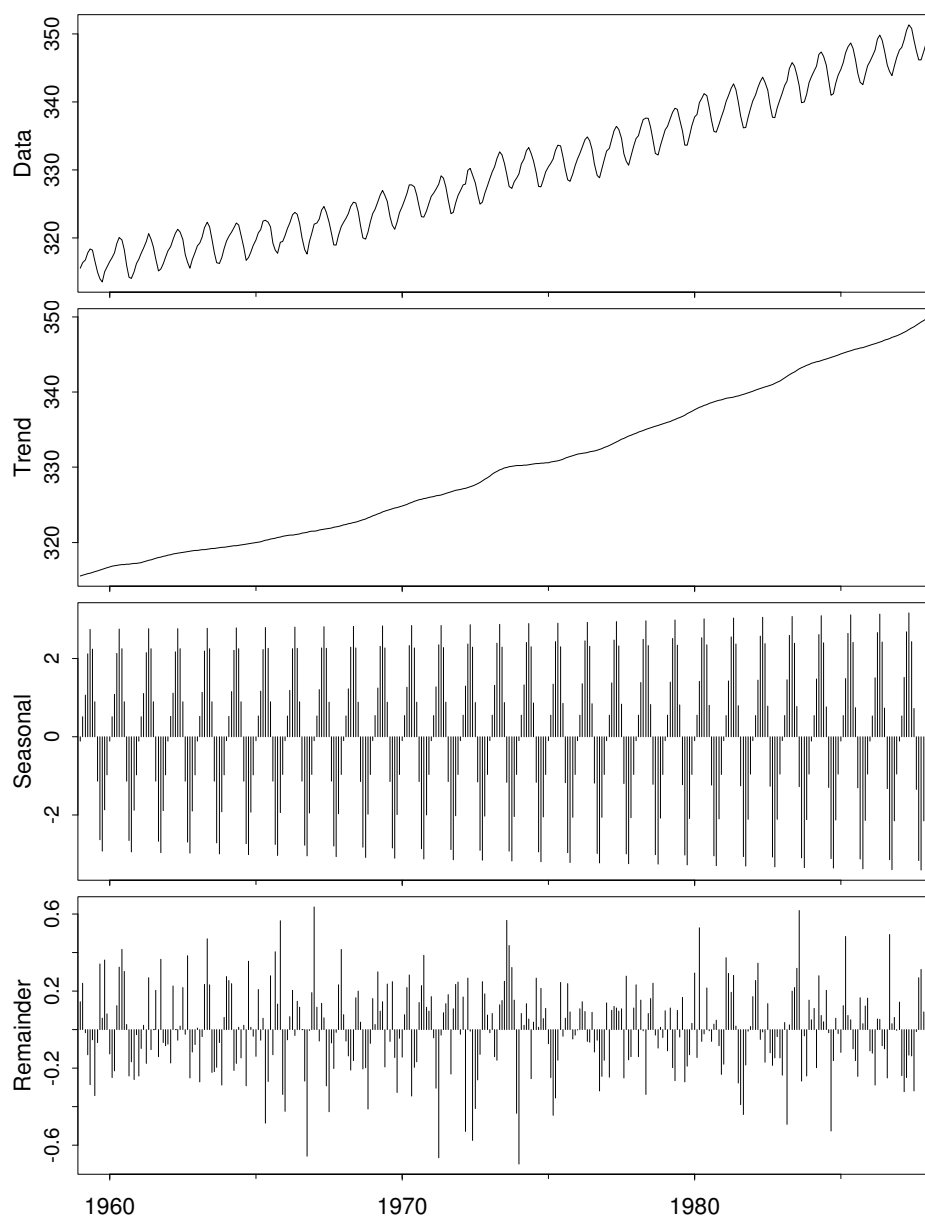
Binomial Log-Likelihood:

$$\ell(p) = \sum_i \{y_i \log p(x_i) + (1 - y_i) \log(1 - p(x_i))\}$$

Other Examples

- poisson regression—
$$\log(\mu(x)) = \alpha + f_1(x_1) + f_2(x_2) + \cdots + f_p(x_p)$$
- discrete choice models — survey response data
- resistant additive models — tapered likelihoods
- semiparametric additive models for designed experiments
- additive decomposition of time series
- additive autoregression models
- varying coefficient models —
$$\eta(x, t) = \alpha(t) + x_1\beta_1(t) + x_2\beta_2(t)$$

Seasonal-Trend Decomposition



Data are monthly CO_2 measurements recorded at Mauna-Loa, Hawaii.

Fitting Additive Models

$$Y = f_1(x_1) + f_2(x_2) + \cdots + f_p(x_p) + \varepsilon$$

Backfitting Equations

$$f_1(x_1) = S_1(y - \bullet - f_2(x_2) - \cdots - f_p(x_p))$$

$$f_2(x_2) = S_2(y - f_1(x_1) - \bullet - \cdots - f_p(x_p))$$

$$\vdots$$

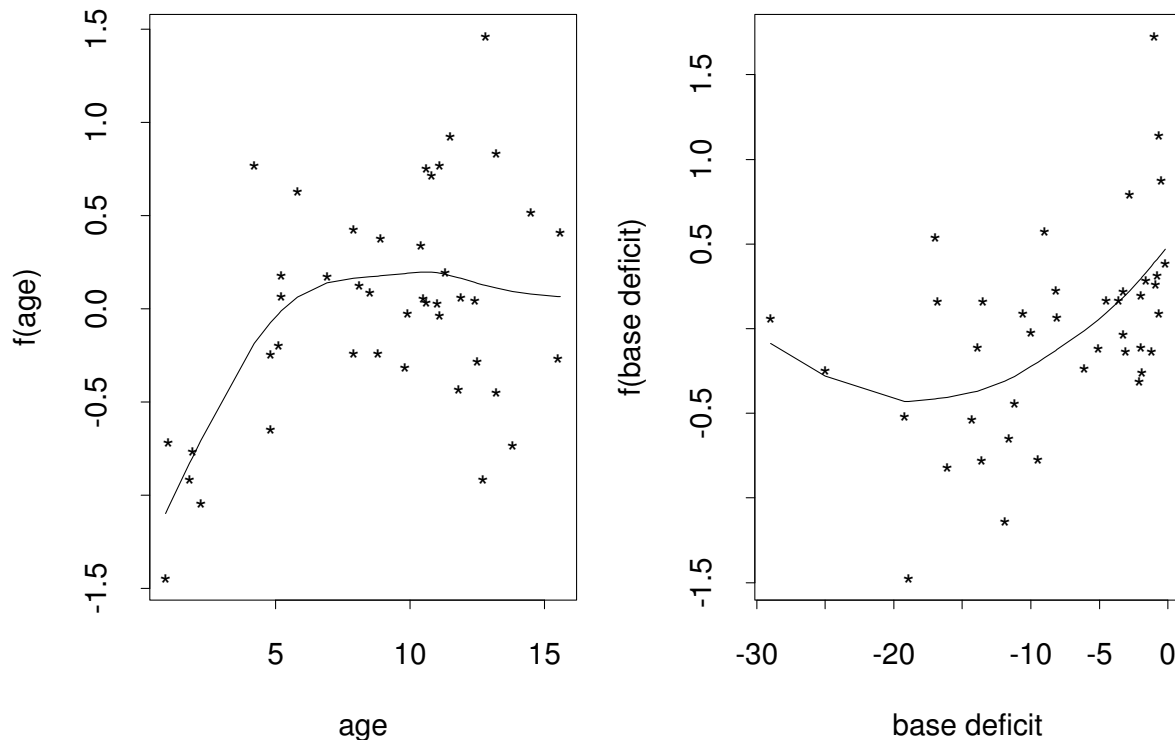
$$f_p(x_p) = S_p(y - f_1(x_1) - f_2(x_2) - \cdots - \bullet)$$

where S_j are:

- linear regression operators yielding polynomial fits, piecewise constant fits, parametric spline fits, etc
- univariate regression smoothers such as smoothing splines, lowess, kernel
- more complicated operators such as surface smoothers for 2nd order interactions

Iteration of smoothing steps above is called the Gauss-Seidel or “backfitting” algorithm.

Smoothing of partial residuals



When fitting the additive model

$$\log(\text{C-peptide}) \sim f_1(\text{age}) + f_2(\text{base deficit})$$

we alternate the two backfitting steps

$$f_1(\text{age}) = \mathbf{S}(\log(\text{C-peptide}) - f_2(\text{base deficit}) | \text{age})$$

$$f_2(\text{base deficit}) = \mathbf{S}(\log(\text{C-peptide}) - f_1(\text{age}) | \text{base deficit})$$

Justification?

Example: penalized least squares

$$\min_{\{f_j\}_1^p} \sum_{i=1}^n \left(y_i - \sum_{j=1}^p f_j(x_{ij}) \right)^2 + \sum_{j=1}^p \lambda_j \int (f_j''(t))^2 dt$$

\Updownarrow

$$f_1 = S_1(\lambda_1)(y - \sum_{j \neq 1} f_j)$$

$$f_2 = S_2(\lambda_2)(y - \sum_{j \neq 2} f_j)$$

\vdots

$$f_p = S_p(\lambda_p)(y - \sum_{j \neq p} f_j)$$

where $S_j(\lambda_j)$ denotes a smoothing spline using variable x_j and penalty coefficient λ_j . Backfitting converges to the minimizer.

Fitting generalized additive models

Logistic Regression

- Compute starting values: f_j^{old} and $\eta^{old} = \sum_j f_j^{old}(x_j)$ e.g. using linear logistic regression
- Iterate
 - construct adjusted dependent variable

$$z_i = \eta_i^{old} + \frac{(y_i - p_i^{old})}{p_i^{old}(1 - p_i^{old})}$$

- construct weights $w_i = p_i^{old}(1 - p_i^{old})$
 - compute $\eta^{new} = A_w z$, the weighted additive model fit to z .
- Stop when functions don't change

This is a Newton-Raphson algorithm for a penalized log-likelihood problem.


Example: Predicting e-mail spam

- data from 4601 email messages
- goal: predict whether an email message is **spam** (junk email) or good.
- input features: relative frequencies in a message of 57 of the most commonly occurring words and punctuation marks in all the training the email messages.
- for this problem not all errors are equal; we want to avoid filtering out good email, while letting spam get through is not desirable but less serious in its consequences.
- we coded **spam** as 1 and **email** as 0.
- A system like this would be trained for each user separately (e.g. their word lists would be different)

Predictors

- 48 quantitative predictors—the percentage of words in the email that match a given word. Examples include `business`, `address`, `internet`, `free`, and `george`. The idea was that these could be customized for individual users.
- 6 quantitative predictors—the percentage of characters in the email that match a given character. The characters are `ch;`, `ch(`, `ch[`, `ch!`, `ch$`, and `ch#`.
- The average length of uninterrupted sequences of capital letters: `CAPAVE`.
- The length of the longest uninterrupted sequence of capital letters: `CAPMAX`.
- The sum of the length of uninterrupted sequences of capital letters: `CAPTOT`.

Details

- A test set of size 1536 was randomly chosen, leaving 3065 observations in the training set.
- A generalized additive model was fit, using a cubic smoothing spline with a nominal four degrees of freedom for each predictor.
- What this means is that for each predictor X_j , the smoothing-spline parameter λ_j was chosen so that $\text{trace}[\mathbf{S}_j(\lambda_j)] - 1 = 4$, where $\mathbf{S}_j(\lambda)$ is the smoothing spline operator matrix constructed using the observed values x_{ij} .
 - This is a convenient way of specifying the amount of smoothing in such a complex model. See  pp 129.
 - “Nominal” because the Df depend on the weights of the converged GAM, which are not known in advance.
 - $\text{trace}[\mathbf{S}_j(\lambda_j)] - 1$ because each smoother fits an intercept, but we only need one.

Some important features

39% of the training data were spam.

Average percentage of words or characters in an email message equal to the indicated word or character. We have chosen the words and characters showing the largest difference between [spam](#) and [email](#).

	george	you	your	hp	free	hpl
spam	0.00	2.26	1.38	0.02	0.52	0.01
email	1.27	1.27	0.44	0.90	0.07	0.43

	!	our	re	edu	remove
spam	0.51	0.51	0.13	0.01	0.28
email	0.11	0.18	0.42	0.29	0.01

Results

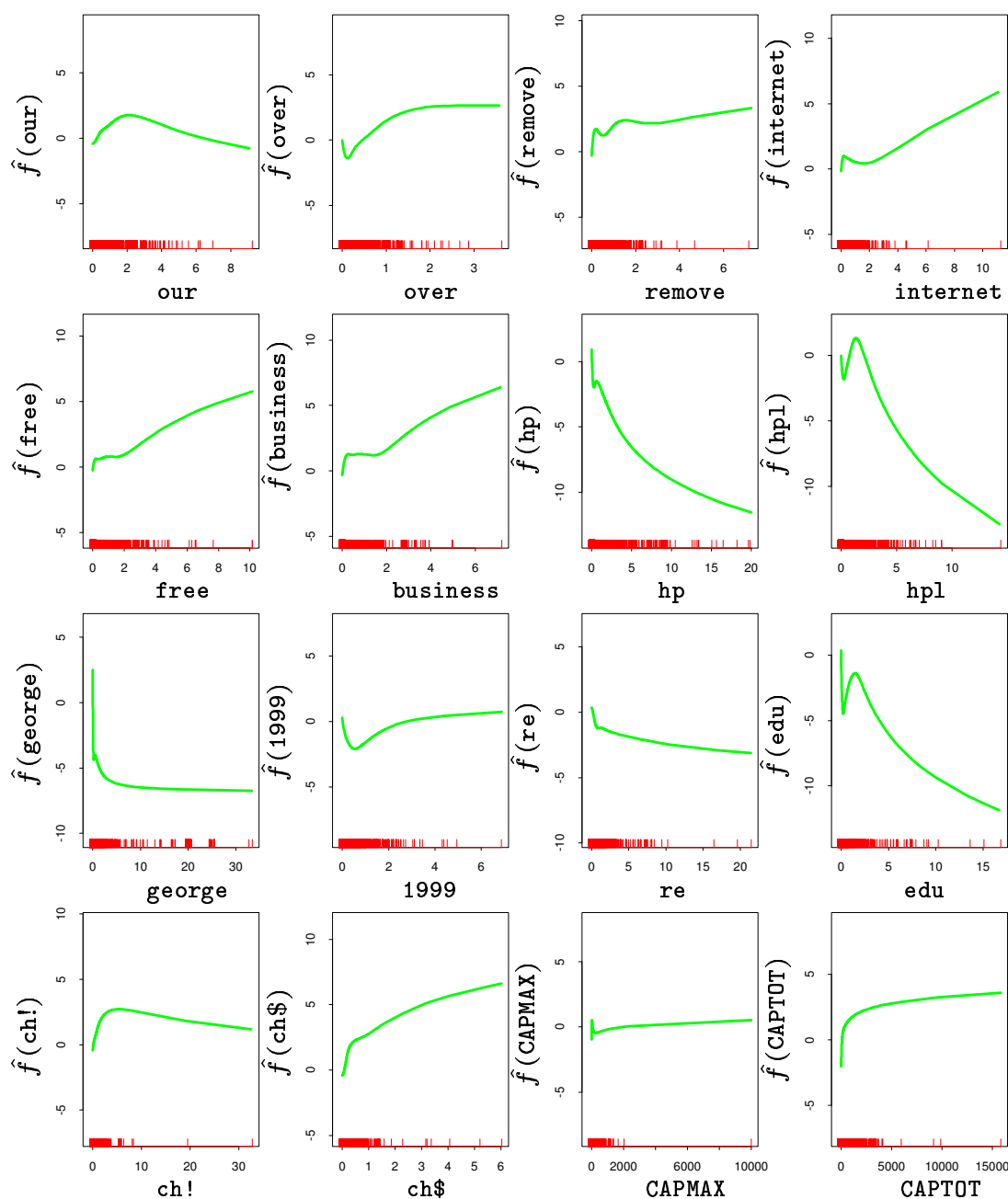
Test data confusion matrix for the additive logistic regression model fit to the spam training data. The overall test error rate is 5.3%.

True Class	Predicted Class	
	<code>email</code> (0)	<code>spam</code> (1)
<code>email</code> (0)	58.5%	2.5%
<code>spam</code> (1)	2.7%	36.2%

Summary of additive logistic fit

Significant predictors from the additive model fit to the spam training data. The coefficients represent the linear part of \hat{f}_j , along with their standard errors and Z-score. The nonlinear p-value represents a test of nonlinearity of \hat{f}_j .

Name	Num.	df	Coef	Std. Err	Z Score	Nonlin P-val
Positive effects						
our	6	3.9	0.566	0.114	4.970	0.052
over	7	3.9	0.244	0.195	1.249	0.004
remove	8	4.0	0.949	0.183	5.201	0.093
internet	9	4.0	0.524	0.176	2.974	0.028
free	17	3.9	0.507	0.127	4.010	0.065
business	18	3.8	0.779	0.186	4.179	0.194
hpl	27	3.8	0.045	0.250	0.181	0.002
ch!	53	4.0	0.674	0.128	5.283	0.164
ch\$	54	3.9	1.419	0.280	5.062	0.354
CAPMAX	57	3.8	0.247	0.228	1.080	0.000
CAPTOT	58	4.0	0.755	0.165	4.566	0.063
Negative effects						
hp	26	3.9	-1.404	0.224	-6.262	0.140
george	28	3.7	-5.003	0.744	-6.722	0.045
1999	38	3.8	-0.672	0.191	-3.512	0.011
re	46	3.9	-0.620	0.133	-4.649	0.597
edu	47	4.0	-1.183	0.209	-5.647	0.000



Spam analysis: estimated functions for significant predictors.

The **rug plot** along the bottom of each frame indicate the observed values of the corresponding predictor. For many predictors the nonlinearity picks up the discontinuity at zero.

Fitting GAMs in Splus

The Splus/R language and environment for data analysis, modeling and graphics comes with tools for fitting

- Linear and anova models
- Generalized linear and **generalized additive models**
- Local regression
- Tree based models
- General nonlinear models
- Time series and spatial models
- Survival models

In addition, a wide variety of contributed software is available for fitting neural networks models, nearest neighbor methods, discriminant analysis, and almost any useful models used in practice.

What tools are available?

Besides a rich functional programming language for graphics and statistical computing, special tools are available for modeling:

- Symbolic formula language
- Data frames
- Software (`lm()`, `aov()`, `glm()`, `gam()` , ...)
- Classes and Methods. For example `plot()`, `print()`, `summary()`, `step()`.

GAMs and Formulas

`gam(NOx ~ C + s(E))`

- reads `NOx` is modeled as `C + s(E)`
- the first term is linear in `C`
- the second term is nonparametric in `E`, to be fit using a smoothing spline.

`NOx ~ s(C) + s(E, df=6)`

`NOx ~ s(C) + lo(E, degree=2, span=.5)`

`log(NOx) ~ C * poly(E,4)`

`NOx ~ lo(C, E)`

Each term in a formula `y ~ a + b` can refer to:

- numeric vector
- factor or logical vector
- matrix
- expression that evaluates to one of the above

Naive Bayes Models

Suppose we estimate the class densities $f_1(X)$ and $f_2(X)$ for the features in class 1 and 2 respectively.

Bayes Formula tells us how to convert these to class posterior probabilities:

$$\Pr(Y = 1|X) = \frac{f_1(X)\pi_1}{f_1(X)\pi_1 + f_2(X)\pi_2},$$

where $\pi_1 = \Pr(Y = 1)$ and $\pi_2 = 1 - \pi_1$.

Since X is often high dimensional, the following **independence** model is convenient:

$$f_j(X) \approx \prod_{m=1}^p f_{jm}(X_m)$$

Works for more than two classes as well.

Naive Bayes continued

- Each of the component densities f_{jm} are estimated separately within each class:
 - Discrete components via histograms
 - quantitative components via Gaussians or smooth density estimates.
- The PAM model has this structure, and in addition
 - assumes the gaussian densities have the same variance in each class
 - shrinks the class centroids towards the overall mean in each class
- More general models have less bias but are typically hard to estimate in high dimensions, so the independence assumption may not hurt too much.

Naive Bayes and GAMs

Note that

$$\log \frac{f_1(X)\pi_1}{f_2(X)\pi_2} = \alpha + \sum_{m=1}^p g_m(X_m),$$

a generalized additive logistic regression model.

GAMs are fit by **binomial maximum likelihood**.

Naive Bayes models are fit using the **full likelihood**.