Generalized Additive Models

September 10, 2019

Motto

My nature is to be linear, and when I'm not, I feel really proud of myself.

Cynthia Weil – a songwriter

Email spam – classification problem

Statistical learning/data mining nomenclature:

- **Training, validating, testing data**: Total available data: 4601 email messages, the true outcome (email type): email or spam is available, along with the relative frequencies of 57 of the most commonly occurring words and punctuation marks.

  In the data mining/big data approach we divide the data into three groups

  - Training data – a half or more of the data
  - Validating data – approximately a half of the remaining data
  - Testing data – the rest of the data
- **Objective**: automatic spam detector – predicting whether the email was junk email
- **Supervised problem**: the outcome is the class (categorical) variable `email/spam`.
- **Classification problem**: the outcomes are discrete (bi-) valued

Features, i.e. predictors

What could be used to predict the outcome? Suggestions?

- 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`.

Statistical Learning Framework

- Data rich situation – we can afford a lot of data
  - **Model fitting – Training set**
  - **Model selection – Validation set (tuning some parameters of the fit or choosing between different models)** [1]
  - **Model assessment – Testing set for the model that was decided to yield the best prediction rate**
- **Training set**: 3065 observations (messages) – the method will be based on these observations
- **Test set**: 1536 messages randomly chosen – the method will be tested on these observation
- In this example there is **no validation** set since the cross-validation approach will be used instead.

[1] This part is often replaced by the cross-validation approach that will be discussed later.

Formalization of the problem

- Coded: `spam` as 'one' and `email` as 'zero'
- $p = 57$ – the number of predictors
- $X_1, \ldots, X_p$ – the predictors themselves
- $\mathcal{X}$ – the space of possible values for predictors, i.e. $(X_1, \ldots, X_p) \in \mathcal{X}$
- Main Task: Divide $\mathcal{X}$ into two disjoint sets $\mathcal{X}_0$ and $\mathcal{X}_1$ and if $(X_1, \ldots, X_p) \in \mathcal{X}_0$ clasify it as `email`, otherwise it is a `spam`.
- How to divide? – Ideas

Conceptual framework

- Suppose that for each randomly selected e-mail message there is a probability that it is a spam.
- Define a **random variable** Y that takes value **1** in the case, when a selected message is a spam and **0** otherwise
- For each randomly chosen message we observe value of **predictors** $X = (X_1, \ldots, X_p)$. They are also **random**.
- The model is completely described by the **joint distribution** of (Y, X). But since X is observable, we are interested only in the **conditional distribution** of Y given X, which is given by

$$P(x) = P(Y = 1 | X = x),$$

i.e. by the probability that a message is a `spam`, given that it is characterized by $X = x$.

Measuring quality of classification

How can we measure the quality a classification method?

- One way is to require that we want very little spam to not be detected.
- A simple rule that every message is a spam would detect all spams but the method is not good – no messages anymore!
- Relaxing the strict requirement, we may look only at the methods that will not detect at most $\alpha 100\%$ spams.
- Among those methods we would like to choose the one that has the smallest percentage of good messages to be classified as spams.
- Finally, and probably most appropriately, we can reverse the role of spam and proper e-mail, i.e. set a strict requirement for the small percentage of e-mail $\alpha 100\%$ to be classified as spam and among methods satisfying it, we would prefer the one that has the smallest percentage of misclassified spams.

Misclassfication rates

- In our probabilistic setup, the chances (percentages) that a regular email is classified as a spam are

$$\alpha = P(X \in \mathcal{X}_1 | Y = 0)$$

while the chances that a spam message is classified as e-mail

$$\bar{\beta} = P(X \in \mathcal{X}_0 | Y = 1)$$

- These two numbers, α and $\bar{\beta}$ are the important characterizations of the classification method given by $\mathcal{X}_0$. We want them to be as small as possible.
- By the **Bayes theorem**[2]

$$P(X \in \mathcal{X}_1 | Y = 0) = \frac{P(Y = 0 | X \in \mathcal{X}_1) P(X \in \mathcal{X}_1)}{P(Y = 0 | X \in \mathcal{X}_1) P(X \in \mathcal{X}_1) + P(Y = 0 | X \in \mathcal{X}_0) P(X \in \mathcal{X}_0)}$$

$$P(X \in \mathcal{X}_0 | Y = 1) = \frac{P(Y = 1 | X \in \mathcal{X}_0) P(X \in \mathcal{X}_0)}{P(Y = 1 | X \in \mathcal{X}_0) P(X \in \mathcal{X}_0) + P(Y = 1 | X \in \mathcal{X}_1) P(X \in \mathcal{X}_1)}$$

[2] Review the concept of conditional probabilities, the total probability formula, and the Bayes theorem!

Estimate $P(X_1, \ldots, X_p)$

- We have seen for the proper analysis of the methods one needs the probability $P(\mathbf{x})$ of spam given $\mathbf{X} = \mathbf{x}$. For example in the Bayes theorem, we have $P(Y = 1 | X \in \mathcal{X}_0)$ and simple property of the conditional probabilities yields

$$P(Y = 1 | X \in \mathcal{X}_0) = E\left(P(X)\right),$$

  where $E(\cdot)$ stands for an expectation of a random variable.
- The main objective now is to find (**estimate**) $P(X_1, \ldots, X_p)$.
- **How?** – Any ideas?
- A simplistic way of doing this:
  - Take all the predictors $(X_1, \ldots, X_p)$ in the training sample and compute frequencies

$$\widehat{P}(X_1, \ldots, X_p) = \frac{\text{\# of times the predictor yields } \texttt{spam}}{\text{\# of times the predictor occurs in the training sample}}$$

There is a problem

- The training sample may not have all possible values in the predictor value space $\mathcal{X}$
- Even for these values that are present in the sample it maybe too few values to get accurate estimate.
- For these reasons our estimate maybe very un-smooth.
- Smoothing methods are needed.

Additive Logistic Regression

- The email spam example is a classification problem that is frequently encountered in a variety of situations
- The additive logistic regression is the model of choice – very popular in medical sciences ('one' can represent death or relapse of a disease).
- $Y = 1$ or $Y = 0$ – a binary variable (outcome)
- $X = (X_1, \ldots, X_p)$ – predictor, features
- A simple but non-linear in X_j's model for the **logit function**

$$\log \frac{P(Y = 1|X)}{P(Y = 0|X)} = \alpha + f_1(X_1) + \cdots + f_p(X_p)$$

Problem is reduced to estimation of α, f_i's

Terminology

- We call the model

$$\log \frac{P(Y=1|X)}{P(Y=0|X)} = \alpha + f_1(X_1) + \cdots + f_p(X_p)$$

  additive because each predictor X_i enters the model individually through adding function $f_i(X_i)$. No interaction terms such as $f(X_1, X_2)$, which would indicate some interaction between feature X_1 and X_2.
- The model will be called **logistic regression** if each of f_i is linear function of X_i, i.e. $f_i(X_i) = \beta_i X_i$.
- In additive logistic regression no parametric form is assumed for f_i.
- One can consider other than linear parametric models, and one can mix various parametric models with non-parametric.

How to connect model with the data?

- The data have the form

$$(y_i, x_{i1}, \ldots x_{ip}),$$

  where the index i runs through samples (e-mail messages in our example).
- The additive logistic regression is written as

$$\log \frac{P(Y=1|X)}{P(Y=0|X)} = \alpha + f_1(X_1) + \cdots + f_p(X_p)$$

- How to connect the two to make a fit?
- Through the **likelihood**!

Binomial model for response

- It is easy to notice the following equivalent formulation of the additive logistic regression model

$$\frac{P(Y=1|\mathbf{X})}{1-P(Y=1|\mathbf{X})} = e^{\alpha+f_1(X_1)+\cdots+f_p(X_p)}$$

$$p(\mathbf{X}) = P(Y=1|\mathbf{X}) = \frac{e^{\alpha+f_1(X_1)+\cdots+f_p(X_p)}}{1+e^{\alpha+f_1(X_1)+\cdots+f_p(X_p)}}$$

- Model for the likelihood: If $(y_1,\ldots,y_N)$ are the observed 0-1 outcomes, corresponding to $(\mathbf{x}_1,\ldots,\mathbf{x}_N)$, the likelihood is

$$\prod_{i=1}^{N} p_{\mathbf{x}_i}^{y_i}(1-p_{\mathbf{x}_i})^{1-y_i}$$

where $p_{\mathbf{x}} = p(\mathbf{x})$. Thus log-likelihood is

$$\sum_{i=1}^{N} y_i(\alpha+f_1(X_{i1})+\cdots+f_p(X_{ip})) - \log(1+e^{\alpha+f_1(X_{i1})+\cdots+f_p(X_{ip})})$$

Maximizing likelihood in linear case

- The log-likelihood function in the classical (linear) logistic regression case is

$$\ell(\alpha, \beta) = \sum y_i(\alpha + \beta_1 X_{i1} + \cdots + \beta_p X_{ip})) - \log(1 + e^{\alpha+\beta_1 X_{i1}+\cdots+\beta_p X_{ip}})$$

- The function is non-linear in α and β's despite it the logit function was linear function of them.
- The first and the second derivatives are easily computable and application of the Newton-Raphson algorithm that uses quadratic approximations can be utilized for computation of the maximum and the resulting MLE $\hat{\alpha}$ and $\hat{\beta}_j$, $j = 1, \ldots, p$.

Newton-Raphson method – basic ideas

- Named after Isaac Newton and Joseph Raphson
- Finding successively better approximations to zeros of a real-valued function f
- We begin with a first guess x_0 for a root of the function f.
- A better approximation x_1 is

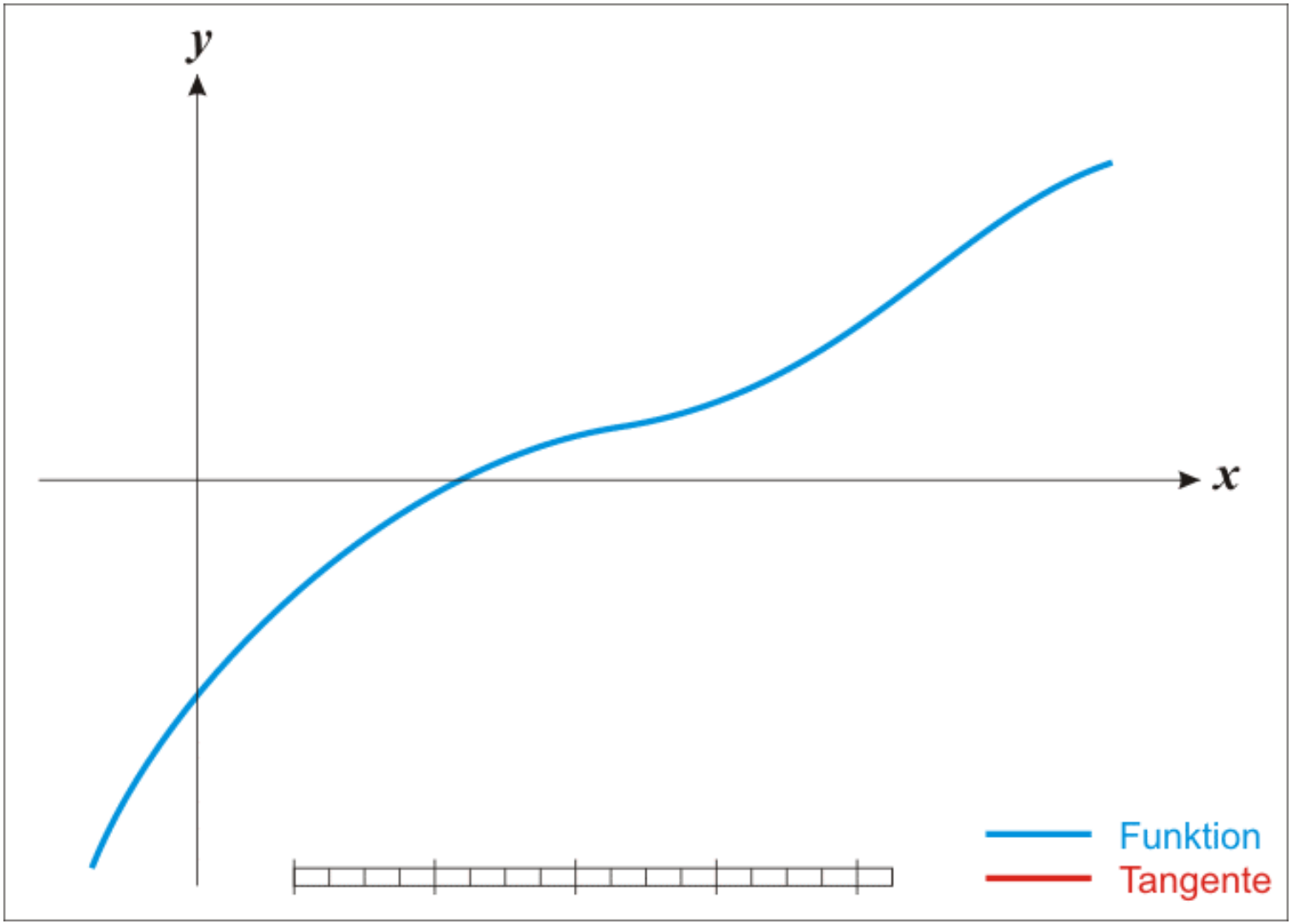
$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}\,.$$

- Geometrically, $(x_1, 0)$ is the intersection with the x-axis of the tangent to the graph at $(x_0, f(x_0))$.
- The process is repeated as

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

until a sufficiently accurate value is reached.

A picture is worth thousand words



Some calculus formulas for our likelihood

- To maximize the log-likelihood will require the derivative and the second derivatives of the likelihood. They can be obtained by application basic multivariate calculus. We report the results without showing (simple) derivations (see also Assignment 3).
- The **first derivatives**

$$\frac{\partial \ell}{\partial \alpha} = \sum_{i=1}^{N} (y_i - p(\mathbf{x}_i, \alpha, \beta)),$$

$$\frac{\partial \ell}{\partial \beta_j} = \sum_{i=1}^{N} x_{ij}(y_i - p(\mathbf{x}_i, \alpha, \beta)), \quad j = 1, \ldots, p$$

- The N-R algorithm requires also the **second-derivatives** that constitute the **Hessian matrix**

$$\frac{\partial^2 \ell(\alpha, \beta)}{\partial(\alpha, \beta)\partial(\alpha, \beta)^T} = -\sum_{i=1}^{N} \mathbf{x}_i \mathbf{x}_i^T p(\mathbf{x}_i; \alpha, \beta)(1 - p(\mathbf{x}_i; \alpha, \beta)).$$

Score equations

- To maximize the log-likelihood, we set its derivatives to zero

$$\frac{\partial \ell}{\partial \alpha} = \sum_{i=1}^{N} (y_i - p(\mathbf{x}_i, \alpha, \beta)) = 0,$$

$$\frac{\partial \ell}{\partial \beta_j} = \sum_{i=1}^{N} x_{ij}(y_i - p(\mathbf{x}_i, \alpha, \beta)) = 0, \quad j = 1, \ldots, p$$

  which are $p + 1$ equations nonlinear in α and β_j's.
- The first score equation specifies that

$$\sum_{i=1}^{N} y_i = \sum_{i=1}^{N} p(\mathbf{x}_i, \alpha, \beta),$$

  i.e. the expected number of 'ones' matches their observed number.
- The Newton-Raphson algorithm requires the second-derivative or Hessian matrix

$$\frac{\partial^2 \ell(\alpha, \beta)}{\partial(\alpha, \beta)\partial(\alpha, \beta)^T} = -\sum_{i=1}^{N} \mathbf{x}_i \mathbf{x}_i^T p(\mathbf{x}_i; \alpha, \beta)(1 - p(\mathbf{x}_i; \alpha, \beta)).$$

Newton-Raphson method

- Starting with $(\alpha^{old}, \beta^{old})$, a single Newton update is

$$(\alpha^{new}, \beta^{new})) = (\alpha^{old}, \beta^{old}) - \frac{\partial^2 \ell(\alpha^{old}, \beta^{old})}{\partial(\alpha, \beta)\partial(\alpha, \beta)^T}^{-1} \frac{\partial \ell(\alpha^{old}, \beta^{old})}{\partial(\alpha, \beta)}$$

  In the above we see clear analogy with the one dimension version of the method seen in the previous slides.

Summary of the N-R method

- Setting:
  $\mathbf{X}$ the $N \times (p+1)$ matrix of x_i values,
  $\mathbf{p}$ the vector of fitted probabilities with ith element $p(\mathbf{x}_i; \alpha^{old}, \beta^{old})$
  $\mathbf{W}$ a $N \times N$ **diagonal matrix of weights** with the ith diagonal element $p(\mathbf{x}_i; \alpha^{old}, \beta^{old})(1 - p(\mathbf{x}_i; \alpha^{old}, \beta^{old}))$
  we get
  $$(\alpha^{new}, \beta^{new}) = (\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{W}\mathbf{z},$$
  where $\mathbf{z} = \mathbf{X}\beta^{old} + \mathbf{W}^{-1}(\mathbf{y} - \mathbf{p})$.
- We see that this algorithm repeatedly solve the least square problem with weights $\mathbf{W}$.

Iteratively Reweighted Least Squares

Generalized Models for Regression

- Similar approach as was seen in the logistic regression one can apply to general regression model
- Consider an arbitrary typically continuous response variable Y.
- We have p predictors $X_1, \ldots, X_p$ and we want to extend beyond the linear regression model.
- We want non-linear models

$$Y = \alpha + f(X_1, \ldots, X_p) + \epsilon,$$

with f to be estimated.

Generalized additive model – extending beyond linearity

- In the generalized additive model

$$Y = \alpha + f_1(X_1) + \ldots f_p(X_p) + \epsilon,$$

  the functions f_j's are unknown and possibly non linear
- We want an automatic fit of functions f_j
- Observed predictors

$$\mathbb{X} = \left[x_{ij}\right]_{i=1,\ldots,N, j=1,\ldots,p}$$

- Consider prescribed tuning parameters λ_j corresponding to the smoothness of the fit to f_j (higher value of λ_j leads to smoother estimate)

Using splines for the multivariate predictors

- In the generalized additive model we have more than one predictor variables, i.e. we have p predictors $X_1, \ldots, X_p$.
- However we want an automatic fit of the functions f_j, $j = 1, \ldots, p$ in a similar way as we have seen for the cubic spline fitting with one predictor.
- The **additive form** of the dependence allows us utilize the previous penalized sum of square approach.

Penalized sum of squares

- A smooth solution that minimizes

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

- The solution is $\hat{\alpha}=\bar{y}$ and $\hat{f}_j$ such that for each $j=1,\ldots,p$:

$$\sum_{i=1}^{N}\hat{f}_j(x_{ij})=0$$

  and $\hat{f}_j$ are smooth cubic splines with knots at each of x_{ij}, $i=1,\ldots,N$.

- Evaluating smoothing cubic splines was discussed before in the lecture and in the discussion sessions.

Backfitting

- Fitting a model involving **multiple predictors**.
- Repeatedly updating the **fit for each predictor** in turn, holding the **others fixed**.
- Each time we update a function, we simply apply the fitting method for that variable to a **partial residual**.
- A partial residual for X_3 in the model $y_i = f_1(x_{i1}) + f_2(x_{i2}) + f_3(x_{i3}) + \epsilon_i$, for example, has the form $r_i = y_i - f_1(x_{i1}) - f_2(x_{i2})$.
- We treat this residual as a response in a non-linear regression on X_3.
- In the following discussion for the jth predictors: $x_{1j}, \ldots, x_{Nj}$, and the corresponding response values $u_1, \ldots, u_N$ the smoothing cubic spline is denoted by $S_j(u_1, \ldots, u_N)$.

The Backfitting Algorithm for Additive Models

Algorithm 9.1 *The Backfitting Algorithm for Additive Models.*

1. Initialize: $\hat{\alpha} = \frac{1}{N}\sum_1^N y_i$, $\hat{f}_j \equiv 0, \forall i, j$.

2. Cycle: $j = 1, 2, \ldots, p, \ldots, 1, 2, \ldots, p, \ldots,$

$$\hat{f}_j \leftarrow \mathcal{S}_j\left[\{y_i - \hat{\alpha} - \sum_{k \neq j} \hat{f}_k(x_{ik})\}_1^N\right],$$

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

until the functions $\hat{f}_j$ change less than a prespecified threshold.

The second step is taken for stability reasons to assure that

$$\sum_{i=1}^N \hat{f}_j(x_{ij}) = 0$$

Logistic additive regressions – more work

- Fitting functions $f_1, \ldots, f_p$ in the logistic additive model is slightly more challenging than in the regression set-up.
- Smoothing splines can still be used.
- But it will require some modification to the backfitting algorithm.
- It is not very important to know details.
- If one is interested then they can be found in **Hastie, T. and Tibshirani, R. (1990)** *Generalized Additive Models*, **Chapman & Hall, London.**
- We will briefly overview the method.
- Let us start with a recap or smoothing splines.

Smoothing splines – regularizing by a penalty

- Spline basis methods that avoids the knot selection
- It is using the maximal set of knots
- It is not overfitting because of penalizing irregularity
- It is estimated by a linear function outside the range of predictors (smoothing on the boundaries)
- It minimizes the penalized residual sum of squares

$$PRSS(f, \lambda) = \sum_{i=1}^{N} (y_i - f(x_i))^2 + \lambda \int f''(t)^2 dt$$

- $\lambda = 0$: any fit that interpolates data exactly.
- $\lambda = \infty$: the least square fit (second derivative is zero)

Smoothing B-splines

- We fit by the cubic splines (see previous lectures) with the maximal number of knots

$$f(x) = \sum_{j=1}^{N+4} \gamma_j B_j(x) \tag{1}$$

- The solution has the form

$$\hat{\gamma} = \left(\mathbf{B}^T\mathbf{B} + \lambda\mathbf{\Omega}_B\right)^{-1}\mathbf{B}^T\mathbf{y},$$

where

$$\mathbf{\Omega}_B = \left[\int B_i''(t)B_j''(t)\,dt\right]$$

- To see this substitute (2) to the PRSS – it becomes a regular least squares problem

Generalized additive models – summary

- Goal: fitting the generalized additive model

$$Y = \alpha + f_1(X_1) + \ldots f_p(X_p) + \epsilon,$$

with f_i smooth splines with the smoothing parameter λ_i.

- Method: minimizing penalized sum of squares
- Solution: for a single predictor there is an explicit solution

$$f(x) = \sum_{j=1}^{N+4} \hat{\gamma}_j B_j(x), \tag{2}$$

where $B_j(x)$'s are the cubic splines with the maximal number of knots located at the predictor values x_i's,

$$\hat{\gamma} = \left(\mathbf{B}^T\mathbf{B} + \lambda\mathbf{\Omega}_B\right)^{-1}\mathbf{B}^T\mathbf{y},$$

where

$$\mathbf{\Omega}_B = \left[\int B_i''(t)B_j''(t)\,dt\right],\ \mathbf{B} = [B_j(x_i)]$$

Algorithm for solution in the general case

- Goal: fitting the generalized additive model

$$Y = \alpha + f_1(X_1) + \ldots f_p(X_p) + \epsilon,$$

with f_i smooth splines with the smoothing parameter λ_i.

- Method: minimizing penalized sum of squares
- Solution: In a generalized case, apply the backfit algorithm, the key step is finding smoothing spline $\hat{f}_j$ that fits $x_{1j}, ..., x_{Nj}$ to

$$u_1 = y_1 - \bar{y} - \sum_{k \neq j} \hat{f}_k(x_{1k}), ..., u_N = y_N - \bar{y} - \sum_{k \neq j} \hat{f}_k(x_{Nk})$$

(this spline was denoted by $S_j(u_1, ..., u_N)$, or $S_j(\mathbf{u})$, and its argument is x, not shown explicitely), so that $\hat{f}_j = S_j(u_1, \ldots, u_N)$. In the algorithm $\hat{f}_j$'s are recycled until convergence.

- The smoothed spline S_j is computed the same as before in the one predictor case except now $\mathbf{y}$ is replaced by $\mathbf{u}$, and x_i becomes x_{ij}.

Generalized additive logistic regression

- Goal: Fitting a simple but non-linear in X_j's model for the logit function

$$\log \frac{P(Y=1|X)}{P(Y=0|X)} = \alpha + f_1(X_1) + \cdots + f_p(X_p)$$

using smoothed splines

- There are no explicit 'responses' in this case, i.e. the left hand side of the above. But there is likelihood:

$$\prod_{i=1}^{N} p_{\mathbf{x}_i}^{y_i}(1-p_{\mathbf{x}_i})^{1-y_i}$$

and the log-likelihood is

$$\sum_{i=1}^{N} y_i(\alpha + f_1(X_{i1}) + \cdots + f_p(X_{ip})) - \log(1 + e^{\alpha + f_1(X_{i1}) + \cdots + f_p(X_{ip})})$$

Maximizing the penalized log-likelihood

- The log-likelihood is non-linear

$$\sum_{i=1}^{N} y_i(\alpha + f_1(X_{i1}) + \cdots + f_p(X_{ip})) - \log(1 + e^{\alpha + f_1(X_{i1}) + \cdots + f_p(X_{ip})})$$

- In analogous approach to the penalized least squares, we can maximize it with the penalty term

$$\sum_{i=1}^{N} y_i(\alpha + f_1(X_{i1}) + \cdots + f_p(X_{ip})) - \log(1 + e^{\alpha + f_1(X_{i1}) + \cdots + f_p(X_{ip})}) - \\ - \sum_{j=1}^{p} \lambda_j \int f_j''(t)^2 \, dt$$

- The solution is obtained by combination of the backfit algorithm with the Newton-Raphson method of maximizing the likelihood.
- The resulting algorithm is referred to as the local scoring algorithm (see Algorithm 9.2 in Textbook II).

The local scoring algorithm

Algorithm 9.2 *Local Scoring Algorithm for the Additive Logistic Regression Model.*

1. Compute starting values: $\hat{\alpha} = \log[\bar{y}/(1-\bar{y})]$, where $\bar{y} = \text{ave}(y_i)$, the sample proportion of ones, and set $\hat{f}_j \equiv 0 \; \forall j$.

2. Define $\hat{\eta}_i = \hat{\alpha} + \sum_j \hat{f}_j(x_{ij})$ and $\hat{p}_i = 1/[1+\exp(-\hat{\eta}_i)]$.

   Iterate:

   (a) Construct the working target variable

   $$z_i = \hat{\eta}_i + \frac{(y_i - \hat{p}_i)}{\hat{p}_i(1-\hat{p}_i)}.$$

   (b) Construct weights $w_i = \hat{p}_i(1-\hat{p}_i)$

   (c) Fit an additive model to the targets z_i with weights w_i, using a weighted backfitting algorithm. This gives new estimates $\hat{\alpha}, \hat{f}_j, \; \forall j$

3. Continue step 2. until the change in the functions falls below a prespecified threshold.

Example from the textbook – spam data

- We apply a generalized additive model to the spam data.
- The data consists of information from 4601 email messages (random test set of size 1536 the rest is in the training set), in a study to screen email for 'spam' (i.e., junk email coded as one). (The data was donated by George Forman from Hewlett-Packard laboratories, Palo Alto, California – the reason for the counts of `george` as a predictor.)
- After some tweaking the model the fit was made for the generalized additive logistic regression model using a cubic smoothing spline with a nominal four degrees of freedom for each predictor. (i.e. for each predictor X_j, the smoothing-spline parameter λ_j was chosen so that $trace[\mathbf{S}_j(\lambda_j)]1 = 4$, where $\mathbf{S}_j(\lambda)$ is the spline operator matrix constructed using the observed values x_{ij}, $i = 1, \ldots, N$ (a way of specifying the smoothing in such a complex model).
- Most of the spam predictors have a very long-tailed distribution so before fitting the GAM model, we log-transformed each variable (actually $\log(x + 0.1)$), (the plots in Figure 9.1 are in the original variables).

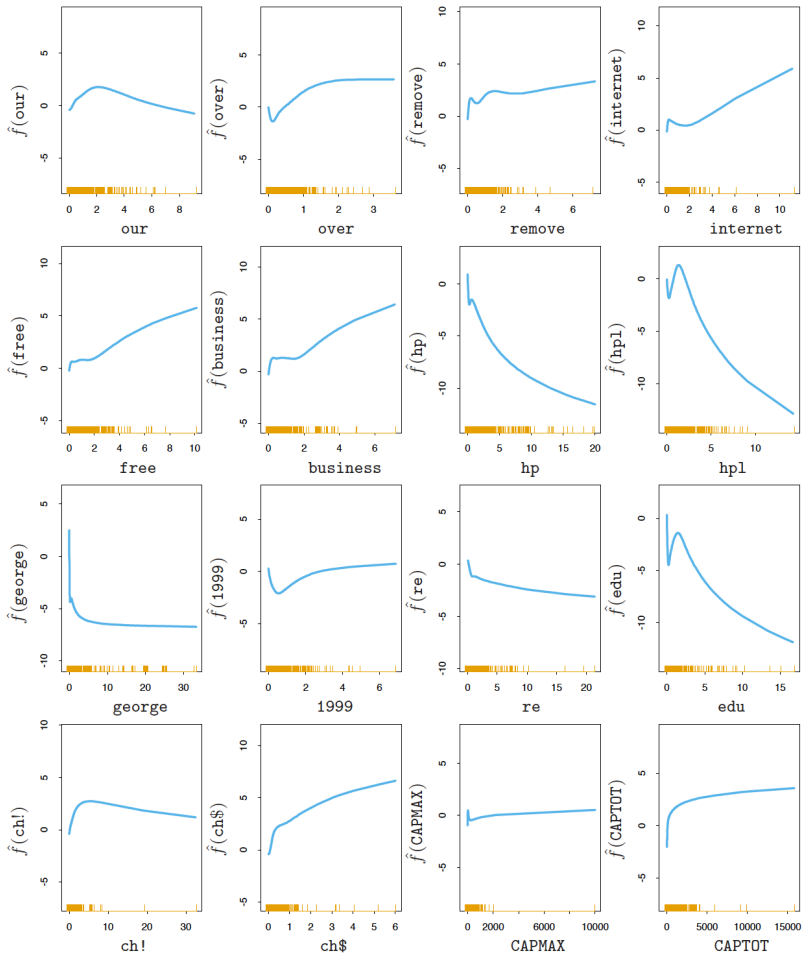


FIGURE 9.1. *Spam analysis: estimated functions for significant predictors. The* rug plot *along the bottom of each frame indicates the observed values of the corresponding predictor. For many of the predictors the nonlinearity picks up the discontinuity at zero.*

Results

- The confusion table of the additive logistic regression fit based on test data set

True Class	Predicted Class	
	email (0)	spam (1)
email (0)	58.3%	2.5%
spam (1)	3.0%	36.3%

- The overall error rate is 5.3%. By comparison, a linear logistic regression has a test error rate of 7.6%.

Results, cont.

- Table 9.2 shows the highly significant predictors.
- For ease of interpretation, the contribution for each variable is decomposed into a linear component and the remaining nonlinear component.
- The top block of predictors are positively correlated with spam, while the bottom block is negatively correlated.
- The linear component is a weighted least squares linear fit of the fitted curve on the predictor, while the nonlinear part is the residual.

TABLE 9.2. *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 is for a test of nonlinearity of $\hat{f}_j$.*

Name	Num.	df	Coefficient	Std. Error	Z Score	Nonlinear P-value
Positive effects						
our	5	3.9	0.566	0.114	4.970	0.052
over	6	3.9	0.244	0.195	1.249	0.004
remove	7	4.0	0.949	0.183	5.201	0.093
internet	8	4.0	0.524	0.176	2.974	0.028
free	16	3.9	0.507	0.127	4.010	0.065
business	17	3.8	0.779	0.186	4.179	0.194
hpl	26	3.8	0.045	0.250	0.181	0.002
ch!	52	4.0	0.674	0.128	5.283	0.164
ch$	53	3.9	1.419	0.280	5.062	0.354
CAPMAX	56	3.8	0.247	0.228	1.080	0.000
CAPTOT	57	4.0	0.755	0.165	4.566	0.063
Negative effects						
hp	25	3.9	−1.404	0.224	−6.262	0.140
george	27	3.7	−5.003	0.744	−6.722	0.045
1999	37	3.8	−0.672	0.191	−3.512	0.011
re	45	3.9	−0.620	0.133	−4.649	0.597
edu	46	4.0	−1.183	0.209	−5.647	0.000