Section 6. The generalized additive model

31 Oct 01

Nonparametric regression

Model $\quad Y = f(X) + \epsilon$

f: smooth $\qquad \epsilon$: noise, mean 0, $\perp f(X)$

$$f(x) = E\{Y \mid X = x\}$$

This conditional expectation has two interpretations:

1) $\min\limits_{\substack{f \\ Ef^2 < \infty}} E\{(Y - f(X))^2\}$

Proof. $E\{(Y - EY|X + EY|X - f(X))^2$

$= E\{(Y - EY|X)^2\} + E\{(EY|X - f(X))^2\}$ $\qquad E = 0$

$+ 2 \underset{X}{E}\{\underset{Y|X}{E}(\quad)(\quad)\}$

2) $\max \operatorname{corr}\{Y, f(X)\}$

This result will be used to motivate ace() later.

D. R. Brillinger
31 Oct 01

Model and large sample properties.

Suppose $Y_1, \ldots, Y_n$ are independent r.v.'s with

$$E\{Y_i \mid x_i\} = s(x_i)$$

$$\mathrm{var}\{Y_i \mid x_i\} = \sigma(x_i)^2$$

Consider

$$\hat{s}(x) = \sum Y_i K_n(x - x_i) / \sum K_n(x - x_i)$$

where $K_n(x) = \frac{1}{b_n} K(x/b_n)$ for some binwidth b_n.

$$E\,\hat{s}(x) = \sum s(x_i) K_n(x - x_i) / \sum K_n(x - x_i)$$

and

$$\mathrm{var}\,\hat{s}(x) = \sum \sigma(x_i)^2 K_n(x - x_i)^2 / \left(\sum K_n(x - x_i)\right)^2$$

Let us look for approximations of the terms appearing, using the Lemma and supposing $b_n \to 0$ as $n \to \infty$.

$$\frac{1}{n} \sum K_n(x - x_i) = \int K_n(x - u) f_X(u)\, du + \text{remainder}$$

$$= \int K(v) f_X(x - b_n v)\, dv + \text{remainder}$$

$$\simeq f_X(x) \int K(v)\, dv, \quad \text{for smooth } f$$

31 Oct 01

Next

$$\frac{1}{n}\sum s(x_i)K_n(x-x_i) = \int s(u)K_n(x-u)f_X(u)\,du + \text{remainder}$$

$$= \int K(v)s(x-b_n v)f_X(x-b_n v)\,dv + \text{remainder}$$

$$\simeq s(x)f(x)\int K(v)\,dv, \text{ for smooth } s\ f$$

So provided $f_X(x), \int K(v)\,dv \neq 0$

$\boxed{E\ \hat{s}(x) \simeq s(x)}$, ie asymptotically unbiased.

Next consider the variance. As before

$$\frac{1}{n}\sum \sigma(x_i)^2 K_n(x-x_i)^2 = \int \sigma(u)^2 K_n(x-u)^2 f_X(u)\,du + \text{remainder}$$

$$= \frac{1}{b_n}\int K(v)^2 \sigma(x-b_n v)^2 f_X(x-b_n v)\,dv + \text{remainder}$$

$$\simeq \frac{1}{b_n}\sigma(x)^2 f_X(x)\int K(v)^2\,dv$$

So the variance

$$\simeq n\frac{1}{b_n}\sigma(x)^2 f_X(x)\int K(v)^2\,dv \Big/ \left(n\ f(x)\int K(v)\,dv\right)^2$$

$$\simeq \boxed{\frac{1}{n b_n}\ \frac{\sigma(x)^2}{f_X(x)}\ \frac{\int K(v)^2\,dv}{\left(\int K(v)\,dv\right)^2}}$$

This will tend to 0 provided $n b_n \to \infty$. Also estimate then consistent.

31 Oct 01

Local weighting / local likelihood.

Model: $f(y|\theta)$

Suppose measurements made in time, (t_i, y_i)

Wish $\hat{\theta}(t)$ as θ may be changing

I. Local likelihood

$$\max_{\theta} \prod_{i: |t_i - t| < b} f(y_i|\theta)$$

II. Local weighting.

$$\max_{\theta} \sum_i k\left(\frac{t-t_i}{b}\right) \log f(y_i|\theta)$$

cp. loess

Tibshirani

4 Mar 01

We have been considering the models

$$Y = s(x) + \text{noise} \qquad s: \text{smooth}$$

$$Y = s(x_1, x_2) + \text{noise} \qquad s: \text{smooth}$$

Now turn to

$$Y = s_1(x_1) + s_2(x_2) + \text{noise} \qquad s: \text{smooth}$$

The generalized additive model (gam)

Hastie and Tibshirani gam( )

Another technique - projection pursuit

5 Mar 01

glm.

Basics. $(\underset{\sim}{x}, Y)$

$$EY = \mu(\underset{\sim}{x})$$

$$\eta = \underset{\sim}{x}'\underset{\sim}{\beta}$$

$$= g(\mu) \quad \text{link function}$$

$$\mu = h(\eta) \quad \text{inverse link}$$

$$\text{var}\, Y = V(\mu)$$

5 Nov 01

to get $\hat{\beta}_1$, then $\hat{\eta}_1$; then $\hat{\mu}_1 = h(\eta_1)$

To get ~~started~~ take $\hat{\mu}_0 = y$, i.e. the response

The setup (*) even suggests large sample distribution.

(1) 6 Nov. 01

Generalized linear model.

Exponential family

$$p_Y(y;\theta,\varphi) = \exp\left\{\frac{y\theta - b(\theta)}{a(\varphi)} + c(y,\varphi)\right\}$$

θ: natural parameter
φ: dispersion parameter

$\mu = E(Y)$ related to covariates $X_1, \dots, X_p$ by

$$g(\mu) = \eta$$

where

$$\eta = \alpha + X_1\beta_1 + \dots + X_p\beta_p$$

is the linear predictor and $g(\cdot)$ is the link function.

$\{\mu = b'(\theta)$, the canonical link$\}$

$$w_i^{-1} = \left(\frac{\partial \eta_i}{\partial \mu_i}\right)_0 V_i^0$$

weights

$$V_i^0 \equiv \text{variance of } Y \text{ at } \mu_i^0$$

Regress $z_i = \eta_i^0 + (y_i - \mu_i^0)\left(\frac{\partial \eta_i}{\partial \mu_i}\right)_0$ on x^i with weights w_i

6 Mar 01

generalized additive variant

Now

$$g(\mu) = \alpha + \sum_{j=1}^{p} f_j(X_j)$$

Local scoring algorithm

1) Initialize.

$$\alpha = g(\bar{y})$$

$$f_1^0, \ldots, f_p^0 = 0$$

2) Update

$$z_i = \eta_i^0 + (y_i - \mu_i^0)\left(\frac{\partial \eta_i}{\partial \mu_i}\right)_0$$

with

$$\eta_i^0 = \alpha^0 + \sum_{j=1}^{p} f_j^0(x_{ij})$$

$$\mu_i^0 = g^{-1}(\eta_i^0)$$

$$w_i = \left(\frac{\partial \mu_i}{\partial \eta_i}\right)_0^2 (V_i^0)^{-1}$$

 6 Nov 01

Fit a <u>weighted additive</u> model to z_i to

obtain: ~~estimated~~ functions f_j^1
predictor η^1
fitted values μ_i^1

Convergence criterion

$$\Delta(\eta^1, \eta^0) = \sum_{j=1}^{p} \| f_j^1 - f_j^0 \| \Big/ \sum_{j=1}^{p} \| f_j^0 \|$$

iii) Repeat step ii) replacing η^0 by η^1 until $\Delta(\eta^1, \eta^0)$ is below some small threshold

6 Nov. 2001

Alternate approach to gam.

Penalized likelihood

linear predictor

$$\eta_i = \alpha + \sum_{j=1}^{p} f_j(x_{ij})$$

log likelihood $\quad l(\underset{\sim}{\eta}; \underset{\sim}{y})$

Find functions $f_1, \ldots, f_p$ to maximize

$$l(\underset{\sim}{\eta}; y) - \frac{1}{2} \sum_{j=1}^{p} \lambda_j \int [f_j''(x)]^2 dx$$

$$\lambda_j \geq 0$$

Cox & O'Sullivan (1985)

6 Nov 2001

<u>Resistant fitting of additive models.</u>

Penalized M-estimate

$$\sum_{i=1}^{n} \hat{\sigma}^2 \rho\left\{ \frac{y_i - \sum_{j=1}^{p} f_j(x_{ij})}{\hat{\sigma}} \right\} + \frac{1}{2} \sum_{j=1}^{p} \lambda_j \int [f_j''(x)]^2 dx$$

Iterative re-weighting $\quad \psi = \rho'$

$$w_i = \frac{\psi(r_i / \hat{\sigma})}{r_i / \hat{\sigma}}$$

$$r_i = y_i - \sum_{j=1}^{p} f_j(x_{ij})$$

$$\hat{\sigma} = \text{med} |r_i| / .67$$

Solution sum of cubic splines
Can use Newton-Raphson

$$f(x) = \alpha + \sum_{j=1}^{k+3} \alpha_j B_j(x) \qquad k \text{ knots}$$

linear parametrization, Easy way to think about it all.

6 Nov. 2001

Resistant fitting of gam's

View deviance contribution $D(y_i; \hat{\mu}_i)$ as analog of $(y_i - \hat{\mu}_i)^2$.

$$g(\mu_i) = \alpha + \sum_j f_j(x_{ij})$$

Re-express $\rho(r)$ as $w(r^2)$

Penalized criterion

$$\sum_{i=1}^{n} w\{D(y_i; \mu_i)\} + \frac{1}{2}\sum_{j=1}^{p} \lambda_j \int [f_j''(x)]^2 dx$$

Express f_j using finite dimensional basis

Use penalized iterative reweighted least squares

May need to use

$$D(y_i; \mu_i)/\hat{\sigma}^2$$

eg. $\hat{\sigma} = \text{med } D(y_i; \hat{\mu}_i)$

6 Nov. 2001

gam(, family = robust(binomial))

c/f. robust version of glm()

Instead of minimizing usual

$$\sum_{i=1}^{n} D(y_i ; \mu_i)$$

minimize

$$D_w = \sum_{i=1}^{n} \hat{\sigma}^2 w. \left(\frac{D(y_i ; \mu_i)}{\hat{\sigma}^2} \right)$$

ϕ : robust estimate of scale

Idea: damp down large contributions

eg.

$$
\begin{aligned}
w(t) &= t & t &\le k^2 \\
&= 2k\sqrt{t} - k^2 & t &> k^2
\end{aligned}
$$

eg. $k = 1.345$

Iterative weights get multiplied by a factor which is ~ 1 for small deviance contributions and gets small for large contributions

```
> robust
function(family = gaussian(), scale = 0, k = 1.345, maxit = 10)
{
        family <- as.family(family)
        weight <- family$weight
        new.exp <- eval(if(scale == 0) substitute(expression({
                        if(iter == 1)
                                robweight <- 1
                        else {
                                if(iter == 2) {
                                  robust.scale <- median(abs(family$deviance(mu,
                                    y, w, T, F)))/0.67
                                  attr(w, "robust") <- c(robust.scale, k)
                                }
                                robust.scale <- attr(w, "robust")[1]
                                robweight <- (k * robust.scale)/abs(family$
                                  deviance(mu, y, w, T, F))
                                robweight <- ifelse(robweight > 1, 1, robweight
                                  )
                        }
                }
                ), list(k = k)) else substitute(expression({
                        robweight <- (k * scale)/abs(family$deviance(mu, y, w,
                                T, F))
                        robweight <- ifelse(robweight > 1, 1, robweight)
                        attr(w, "robust") <- c(scale, k)
                }
                ), list(k = k, scale = scale)))
        dummy <- expression(junk * robweight)
        dummy[[1]][[2]] <- weight[[1]]
        new.exp[[1]][[length(new.exp[[1]]) + 1]] <- dummy[[1]]
        family$weight <- new.exp
        family$deviance <- substitute(function(mu, y, w, residuals = F, robust
                 = T)
        {
                old.deviance <- function(mu, y, w, residuals = F)
                body
                if(!robust)
                        return(old.deviance(mu, y, w, residuals))
                a <- attr(w, "robust")
                if(is.null(a))
                        return(old.deviance(mu, y, w, residuals))
                else {
                        robust.scale <- a[1]
                        k <- a[2] * robust.scale
                        dev <- old.deviance(mu, y, w, T)        #
 # remember if there are prior weights they are included here
                        devtest <- abs(dev) <= k
                        devsq <- dev^2 * devtest + (!devtest) * (2 * k * abs(
                                dev) - k^2)
                        if(residuals)
                                sign(dev) * sqrt(devsq)
                        else sum(devsq)
                }
        }
        , list(body = family$deviance[[5]]))
        family$family["name"] <- paste("Robust", family$family["name"])
        family$initialize <- c(family$initialize, substitute(expression(maxit <-
                nit), list(nit = maxit))[2])
        family
}
```

```
Script started on Mon Oct 22 13:29:34 2001
script_wol.brill% Splus<analfromel
License Warning : S-PLUS license expires Wed Oct 31 23:59:59 2001
S-PLUS : Copyright (c) 1988, 1996 MathSoft, Inc.
S : Copyright AT&T.
Version 3.4 Release 1 for Sun SPARC, SunOS 5.3 : 1996
Working data will be in /saruman/accounts/fac/brill/.Data

Call: glm(formula = y ~ a + b + offset(log(n)), family = "poisson")
Deviance Residuals:
       Min         1Q     Median        3Q      Max
 -1.832904 -0.8559731 -0.3807713 0.4241323 2.176178

Coefficients:
                 Value Std. Error     t value
(Intercept) -7.2810177  0.1724004 -42.2331955
         a1 -2.1787235  0.5027471  -4.3336372
         a2 -0.9586813  0.3663622  -2.6167581
         a3 -0.0796293  0.2562401  -0.3107605
         a4  0.1302123  0.2492969   0.5223182
         a5  0.7221383  0.1716511   4.2070126
         a6  0.9374875  0.1689896   5.5476038
         b1 -3.1187052  0.8933875  -3.4908761
         b2 -2.1717629  0.5299118  -4.0983481
         b3 -1.4171269  0.3886831  -3.6459703
         b4  0.0842177  0.2294316   0.3670711
         b5  0.1235435  0.2421091   0.5102803
         b6  1.0901213  0.2050192   5.3171658
         b7  1.3289269  0.2177310   6.1035266
         b8  1.7861285  0.2292702   7.7904966

(Dispersion Parameter for Poisson family taken to be 1 )

    Null Deviance: 445.099 on 62 degrees of freedom

Residual Deviance: 51.47087 on 48 degrees of freedom

Number of Fisher Scoring Iterations: 5

Call: glm(formula = y ~ d + b + offset(log(n)), family = poisson)
Deviance Residuals:
       Min        1Q     Median        3Q      Max
 -1.991026 -1.202705 -0.3255014 0.4096175 2.018623

Coefficients:
                  Value Std. Error    t value
(Intercept) -11.7739704 0.37019955 -31.804389
          d   0.4886346 0.04956777   9.857911
          b   0.5637749 0.03775633  14.931931

(Dispersion Parameter for Poisson family taken to be 1 )

    Null Deviance: 445.099 on 62 degrees of freedom

Residual Deviance: 71.21102 on 60 degrees of freedom

Number of Fisher Scoring Iterations: 4

Correlation of Coefficients:
  (Intercept)          d
d -0.7462301
b -0.6817992   0.0678269

Call: gam(formula = y ~ lo(d) + lo(b) + offset(log(n)), family = poisson)
Deviance Residuals:
```

British physicians: 1 & 2 fitted factors, 3 & 4 via~~n~~ gam