Section 6. The generalized additive model
$\text{Nonparametric regression.}$

Model: $y = f(x) + \epsilon$

- $f$: smooth
- $\epsilon$: noise, mean $0$, $\perp f(x)$

$f(x) = E\{y | x = x\}$

This conditional expectation has two interpretations:

1) mean $E f(Y | f(X))$

2) max corr $\{y, f(x)\}$

Proof: $E \{ (y - E y | x) + (E y | x) - f(x) \}^2$

$= E \{ (y - E y | x)^2 \} + E \{ (E y | x) - f(x) \}^2$

$+ 2 E \{ E (\_)(\_) \}$

$= 0$

This result will be used to motivate $\text{ACOR}$ later.
Model and large sample properties.

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

\[
\text{E} \{ Y_i | x_i \} = \mu(x_i)
\]

\[
\text{var} \{ Y_i | x_i \} = \sigma^2(x_i)
\]

Consider

\[
\hat{\mu}(x) = \frac{\sum_i Y_i K_m(x - x_i)}{\sum_i K_m(x - x_i)}
\]

where

\[
K_m(x) = \frac{1}{b_m} K \left( \frac{x}{b_m} \right)
\]

for some binwidth \( b_m \).

\[
\text{E} \hat{\mu}(x) = \frac{\sum_i \mu(x_i) K_m(x - x_i)}{\sum_i K_m(x - x_i)}
\]

and

\[
\text{var} \hat{\mu}(x) = \frac{\sum_i \sigma^2(x_i) K_m(x - x_i)^2}{\left( \sum_i K_m(x - x_i) \right)^2} - \frac{1}{\sum_i K_m(x - x_i)}
\]

Let us look for approximations of the terms appearing, using the lemma and supposing \( b_m \to 0 \) as \( n \to \infty \).
\[ \frac{1}{n} \sum d(x_i) K_n(x-x_i) = \int d(u) K_n(x-u) f(x) du + \text{remainder} \]
\[ = \int K_n(u) d(x-b_n u) f(x-b_n u) du + \text{remainder} \]
\[ = d(x) f(x) \int K_n(u) du, \text{ for smooth } f \]

So provided \( f(x) \int K_n(u) du \neq 0 \)

\[ E \Delta(x) = \Delta(x) \] is asymptotically unbiased.

Next consider the variance. As before

\[ \frac{1}{n} \sum \sigma^{-2}(x_i) K_n(x-x_i) = \int \sigma^{-2}(u) K_n(x-u) f(x) du + \text{remainder} \]
\[ = \frac{1}{b_n} \int K_n(u) \sigma^{-2}(x-b_n u) f(x-b_n u) du + \text{remainder} \]
\[ = \frac{1}{b_n} \sigma^2(x) f(x) \int K_n(u) du \]

So the variance

\[ = \frac{1}{n b_n} \sigma^2(x) f(x) \int K_n(u) du \left( \frac{\int \sigma^{-2}(u) K_n(u) du}{\int \sigma^2(u) K_n(u) du} \right)^2 \]

This will tend to 0 provided \( nb_n \to \infty \). Also estimate their consistent.
Local weighting/Local likelihood.

Model: \( f(y|\theta) \)

Suppose measurement made in time \( t_i \), \( (t_i, y_i) \)

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

I. Local likelihood

\[
\max_{\theta} \prod_{i: |t_i-t| \leq b} f(y_i|\theta)
\]

II. Local weighting

\[
\max_{\theta} \sum_{i} k\left(\frac{t-t_i}{b}\right) \log f(y_i|\theta)
\]

cp. does

Tibshirani
We have been considering the models

\[ Y = d(x) + \text{noise} \quad \text{s: smooth} \]

\[ Y = d(x_1, x_2) + \text{noise} \quad \text{s: smooth} \]

Now turn to

\[ Y = d_1(x_1) + d_2(x_2) + \text{noise} \quad \text{s: smooth} \]

The generalized additive model (gam)

Hastie and Tibshirani - gam( )

Another technique - projection pursuit
Given.

Baseline: 

\[(x, y)\]

\[E(Y) = \mu(x)\]

\[n = x' \beta\]

\[= g(\mu)\]

\[\mu = h(n)\]

\[\text{var}(y) = v(\mu)\]
2)  

5 Nov 01

to get $\hat{\beta}_1$, then $\hat{\alpha}_i$, then $\hat{\mu}_i = h(\eta_i)$.

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

The setup (*) even suggests large sample distribution.
Generalized linear model.

Exponential Family

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

\( \theta \): natural parameter

\( \phi \): dispersion parameter

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

\[ g(\mu) = \eta \]

where

\[ \eta = \alpha + X_1 \beta_1 + \cdots + X_p \beta_p \]

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

\[ \left\{ \begin{array}{l}
\mu = b'(\theta), \text{ the canonical link} \quad \gamma \\\n\end{array} \right. \]

\[ w_i^{-1} = \left( \frac{\partial g(\mu)}{\partial \mu} \right)^{-1} \]

weights

\[ V_i^{0} = \text{variance of } Y \text{ at } \mu_i^{0} \]

Regression \( \beta_i = \mu_i^{0} + (y_i - \mu_i^{0}) \left( \frac{\partial g(\mu)}{\partial \mu} \right) \) on \( x_i \) with weights \( w_i \).
generalized additive variant

Now
\[ g(\mu) = \alpha + \sum_{j=1}^{b} f_j(x_j) \]

Local scoring algorithm

1) Initialize
\[ \mu = g^{-1}(\eta) \]
\[ f^0, \ldots, f^0 = 0 \]

2) Update
\[ \beta_i = \eta^0_i + (y_i - \mu^0_i)(\frac{\partial \eta^0_i}{\partial \mu^0_i}) \]

with
\[ \eta^0_i = \alpha^0 + \sum_{j=1}^{b} \beta^0_{i,j} (x_{ij}) \]
\[ \mu^0_i = g^{-1}(\eta^0_i) \]
\[ w_i = \left(\frac{\partial \mu_i}{\partial \eta_i}\right)^2 (v_i)^{-1} \]
Fit a weighted additive model to $z_i$ to obtain estimated functions $\hat{f}_j$ predicted $\hat{y}_i$ fitted values $\hat{m}_i$.

Convergence criterion:

$$\Delta(\hat{y}, y^0) = \frac{\sum_{j=1}^k \sum_{i=1}^n \left( \hat{y}_i - y^0_i \right)^2}{\sum_{j=1}^k \sum_{i=1}^n (y^0_i)^2}$$

iii) Repeat step ii) replacing $y^0$ by $\hat{y}$ until $\Delta(\hat{y}, y^0)$ is below some small threshold.
Alternative approach to game.

Paraclized likelihood

Linear predictor

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

Log likelihood \( l(\eta; y) \)

Find functions \( f_1, \ldots, f_p \) to maximize

\[ l(\eta; y) = \frac{1}{2} \sum_{j=1}^{p} \gamma_j \int f_j''(x) \bar{y}_j^2 \, dx \]

\( \gamma_j > 0 \)

Cox & O'Sullivan (1985)
Resistant fitting of additive models.

Penalized M-estimates

\[ \sum_{i=1}^{n} \frac{b_i - \sum_{j=1}^{p} \beta_j f_j(x_i)}{\delta_i} + \frac{b}{2} \sum_{j=1}^{p} \left[ \int_{\mathbb{R}} f_j''(x)^2 \, dx \right] \]

Iterative re-weighting

\[ s = \rho \]

\[ w_i = \frac{4 (r_i / s)}{r_i / s} \]

\[ r_i = y_i - \sum_{j=1}^{p} \beta_j f_j(x_i) \]

\[ s = \text{med} |r_i| / 0.67 \]

Solution using cubic splines

Can use Newton-Raphson

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

Linear parameterization. Easy way to think about it all.
Resistant fitting of gamma's

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

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

Re-express $\beta(x)$ as $w(x)$

Penalized criterion

$$\sum_{i=1}^{n} w_i D(y_i; \mu_i) + \frac{1}{2} \sum_j \beta_j \int \left[ f_j''(x) \right]^2 dx$$

Expand $f_j$ using finite dimensional basis

Use penalized iterative reweighted least squares

May need to use $D(y_i; \hat{\mu}_i)/\hat{\sigma}^2$

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

df. robust version of glm()

Instead of minimizing usual
\[ I_1(y_i; \phi) \]

minimize
\[ D_\phi = \sum_{i=1}^{n} \sigma^2 w_i \left( \frac{D(y_i; \phi)}{\sigma^2} \right) \]

\[ \phi : \text{robust estimate of scale} \]

Idea: damp down large contributions

\[ w(t) = \begin{cases} t & t \leq k^2 \\ 2k\sqrt{t} - k^2 & t > k^2 \end{cases} \]

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
    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}
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
(Intercept)  1
   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 \( \text{gam} \)