Methods of Applied Statistics I

STA2101H F LEC9101

Week 11

December 1 2021

Today

Start Zoom

1. Upcoming events
2. Recap on GLMs
3. GLM examples
4. case-control studies
5. nonparametric regression

Upcoming

- Monday Dec 6 3.30 Data Science ARES series
  The Rigorous and Human Life of Data Link

Dr. Cecilia Aragon, U Washington

- Friday Dec 10 Toronto Data Workshop Zoom link

Nathan Taback U of T

... Upcoming

- Thursday Dec 2 3.30
  A modern take on Huber regression

  Po-Ling Loh, U Cambridge

Zoom Link

Project

- Part I 3–5 pages, non-technical
  1. a description of the scientific problem of interest
  2. how (and why) the data being analyzed was collected
  3. preliminary description of the data (plots and tables)
  4. non-technical summary for a non-statistician of the analysis and conclusions

- Part II 3–5 pages, technical
  1. models and analysis
  2. summary for a statistician of the analysis and conclusions

- Part III Appendix

  R script or .Rmd file; additional plots; additional analysis; References

- 40 points total

- Part I:
  description of data and scientific problem 5
  suitability of plots and tables 5 — clear, non-technical, concise but thorough
  quality of the presentation 5

- Part II:
  summary of the modelling and methods 5 — justification for choices
  suitability and thoroughness of the analysis 10 — model checks, data checks

- Part III:
  relevance of additional material 5
  complete and reproducible submission 5

Recap: GLMs

- response y_i, covariates x_i^T — $1 \times p$ vector
- $E(y_i) = \mu_i, \quad g(\mu_i) = x_i^T\beta = \eta_i \qquad \text{var}(y_i) = \phi_i V(\mu_i)$ — $V(\cdot)$ variance function
- ϕ_i is either known, or constant, or $\phi_i = a_i\phi$ — a_i known

- inference for β based on likelihood theory:
  $\ell'(\hat{\beta}) = 0, \qquad \widehat{\text{var}}(\hat{\beta}) = \{-\ell''(\hat{\beta})\}^{-1} \qquad (\hat{\beta} - \beta)^T j(\hat{\beta})(\hat{\beta} - \beta) \dot{\sim} \chi^2_p$
  $\hat{\beta}_j \dot{\sim} N(\beta_j, \widehat{\text{var}}(\hat{\beta})_{jj})$
- $w(\beta) = 2\{\ell(\hat{\beta}) - \ell(\beta)\} \dot{\sim} \chi^2_p \qquad 2\{\ell_p(\hat{\beta}_j) - \ell_p(\beta_j)\} \dot{\sim} \chi^2_d$ — profile log-likelihood

- model-checking as with linear models:
  deviance residuals or Pearson residuals, measures of influence — Cook's distance

- model-building as with linear models:
  forward/backward/stepwise, analysis of deviance, AIC, subject-matter knowledge

Iteratively re-weighted least squares

$$\hat{\beta} = (X^T W X)^{-1} X^T W z; \quad z = X\beta + W^{-1}u; \qquad z(\beta) = X\beta + W^{-1}(\beta)u(\beta); X\beta = \eta$$

$$\text{Var}(\hat{\beta}) \doteq (X^T W X)^{-1}$$

W is diagonal

$$W_{ii} = \frac{1}{\phi a_i \{g'(\mu_i)\}^2 V(\mu_i)}$$

$$u_i = \frac{y_i - \mu_i}{\phi a_i g'(\mu_i) V(\mu_i)}$$

initial values?: $\mu_{init} = y, \quad \eta_{init} = X\beta_{init} = g(y)$

SM §10.3.2; ELM §6.2

$$\begin{aligned}
\hat{\beta} &= (X^T W^{-1} X) X^T W z \\
&\text{iteration} \\
\hat{\beta}^{(t+1)} &= (X^T W^{(t)} X)^{-1} X^T W^{(t)} z^{(t)} \qquad z^{(t)} = X\hat{\beta}^{(t)} + W^{-1(t)} u^{(t)}; \\
W^{(t)} &= W(\hat{\beta}^{(t)}), \quad u^{(t)} = u(\hat{\beta}^{(t)})
\end{aligned}$$

At convergence,

$$\begin{aligned}
\hat{\beta} &= (X^T \hat{W} X)^{-1} X^T \hat{z} \\
\widehat{\mathrm{Var}}(\hat{\beta}) &\doteq (X^T \hat{W} X)^{-1}
\end{aligned}$$

W is diagonal

$$W_{ii} = \frac{1}{\phi a_i \{g'(\mu_i)\}^2 V(\mu_i)}, \qquad u_i = \frac{y_i - \mu_i}{\phi a_i g'(\mu_i) V(\mu_i)}$$

... GLMs special sauce

- for Poisson and binomial responses, the saturated model has residual deviance > 0
- this provides a goodness-of-fit test of the regression model

- for normal or gamma responses, $\phi_i = \phi$ is constant;
  it is estimated from the residuals after fitting the regression model Pearson residuals

- quasi- Poisson or binomial introduces an overdispersion parameter that inflates the estimated variance of $\hat{\beta}_j$

- connection to (weighted) least squares enables fast convergence of algorithm, direct generalization of diagnostics

Gamma glm; Binary deviance (HW6); HW7,8

- ELM §7.1: "The canonical parameter is $-1/\mu$, so the canonical link is $\eta = -1/\mu$. However, we typically remove the minus (which is fine provided we take account of this in any derivations) and just use the inverse link"

- $E(y_i) = \mu_i, \qquad \theta_i = -1/\mu_i, \qquad \text{var}(y_i) = \mu_i^2/\nu = \phi/\theta_i^2$
- if $g(\mu_i) = \theta_i$, then $g'(\mu_i) = 1/V(\mu_i) \longrightarrow g'(\mu_i) = 1/\mu_i^2$
- $\sum(y_i - \hat{\mu}_i)x_{ij} = 0, \quad j = 1, \ldots, p$

- if μ_i increases, $E(y_i)$ increases, θ_i increases, seems all consistent

- binary response; deviance = $-2\sum_{i=1}^{n}\{p_i(\hat{\beta})\log[p_i(\hat{\beta})/\{1 - p_i(\hat{\beta})\}] + \log(1 - p_i(\hat{\beta})\}$
  $= \ldots \qquad = \qquad -2\sum[p_i(\hat{\beta})x_i^T\hat{\beta} + \log\{1 - p_i(\hat{\beta})\}]$

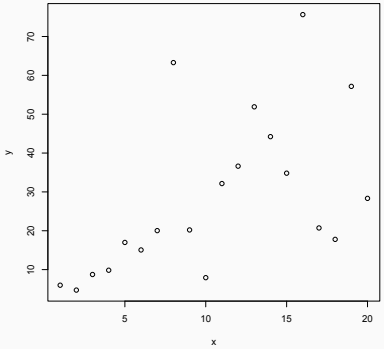
Aside

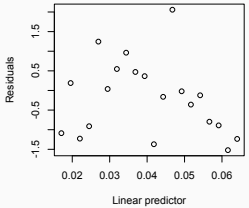
$$
\begin{aligned}
D &= 2\sum\left[y_i \log\left(\frac{y_i}{m_i\hat{p}_i}\right) + (m_i - y_i)\log\left\{\frac{m_i - y_i}{m_i(1-\hat{p}_i)}\right\}\right] \\
&= 2\sum\left\{y_i \log\left(\frac{y_i}{\hat{p}_i}\right) + (1 - y_i)\log\left(\frac{1-y_i}{1-\hat{p}_i}\right)\right\} \\
&= 2\left\{\sum_{y_i=0}\log\left(\frac{1}{1-\hat{p}_i}\right) + \sum_{y_i=1}\log\left(\frac{1}{\hat{p}_i}\right)\right\} \\
&= -2\left\{\sum_{y_i=0}\log(1-\hat{p}_i) + \sum_{y_i=1}\log(\hat{p}_i)\right\} \\
&= -2\left\{\sum_{i=1}^{n}\log(1-\hat{p}_i) + \sum_{y_i=1}\log\left(\frac{\hat{p}_i}{1-\hat{p}_i}\right)\right\} \\
&= -2\left\{\sum_{i=1}^{n}\log(1-\hat{p}_i) + \sum_{i=1}^{n} y_i x_i^T\hat{\beta}\right\} = -2\left\{\sum_{i=1}^{n}\log(1-\hat{p}_i) + \sum_{i=1}^{n}\hat{p}_i x_i^T\hat{\beta}\right\}
\end{aligned}
$$

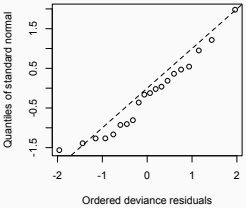
Quick example

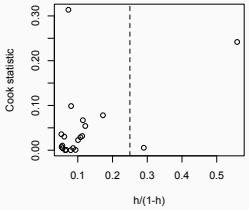
```
n <- 20
b0 <- 3; b1 <- 2
x <- 1:20; y <- x
eta <- b0 + b1*x
mu = 1/eta
for(i in 1:20){y[i] <- rgamma(1,shape = 3, rate = 3/mu[i])}
plot(x,y)

gamglm <- glm(y ~ x, family = Gamma(link = inverse))

summary(gamglm)

plot.glm.diag(gamglm)
```

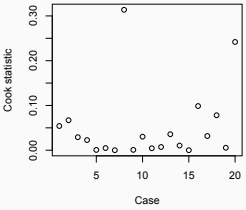
... Quick example











More GLM examples

$\longrightarrow$ glmex.Rmd

see glmex.pdf

Contingency tables

$\longrightarrow$ casecontrol.pdf

two-way tables

Nonlinear least squares

- $y_i = \eta(x_i; \beta) + \epsilon_i$ Example 10.9 $\eta(x_i; \beta) = \beta_0\{1 - \exp(-x_i/\beta_1)\}$
- SM shows how this can also be fit using IRWLS

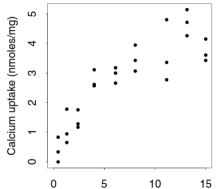
$$\frac{dy}{dx} = (\beta_0 - y)/\beta_1$$

"Allowing for measurement error, which seems to be similar at all levels of y"

Table 10.1 Calcium uptake (nmoles/mg) of cells suspended in a solution of radioactive calcium, as a function of time suspended (minutes) (Rawlings, 1988, p. 403).

Time (minutes)	Calcium uptake (nmoles/mg)		
0.45	0.34170	−0.00438	0.82531
1.30	1.77967	0.95384	0.64080
2.40	1.75136	1.27497	1.17332
4.00	3.12273	2.60958	2.57429
6.10	3.17881	3.00782	2.67061
8.05	3.05959	3.94321	3.43726
11.15	4.80735	3.35583	2.78309
13.15	5.13825	4.70274	4.25702
15.00	3.60407	4.15029	3.42484

Figure 10.1 Calcium uptake (nmoles/mg) of cells suspended in a solution of radioactive calcium, as a function of time suspended (minutes).



- model $y_i = f(x_i) + \epsilon_i, \quad i = 1, \ldots, n \quad x_i$ scalar
- mean function $f(\cdot)$ assumed to be "smooth"
- introduce a kernel function $K(u)$ and define a set of weights

$$w_i = \frac{1}{\lambda} K\left(\frac{x_i - x_0}{\lambda}\right)$$

- estimate of $f(x)$, at $x = x_0$:

$$\hat{f}_\lambda(x_0) = \frac{\sum_{i=1}^n w_i y_i}{\sum_{i=1}^n w_i}$$

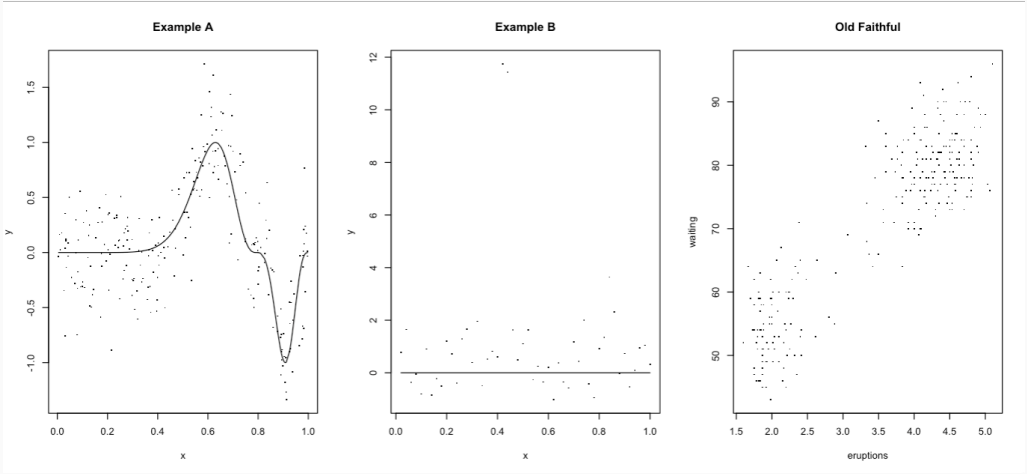
- Nadaraya-Watson estimator – local averaging

local polynomial of degree 0

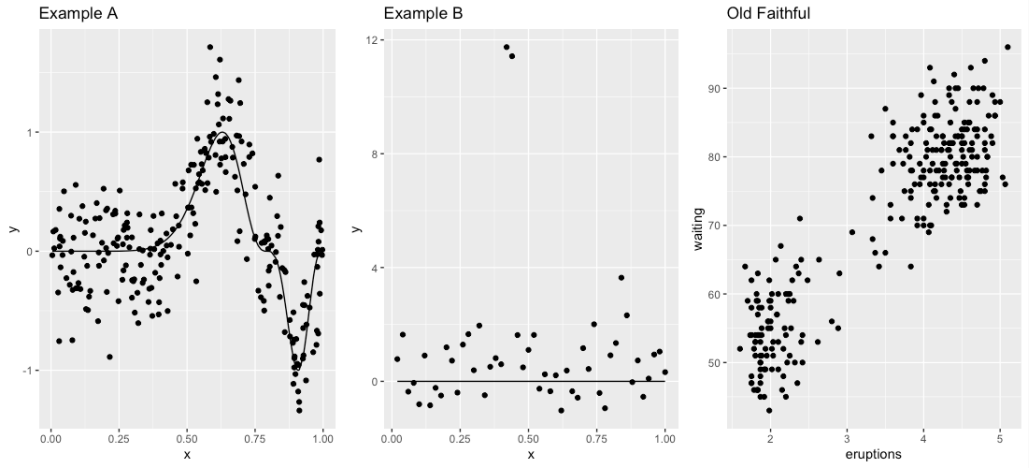
- choice of bandwidth, λ controls smoothness of function
- larger bandwidth $=$ more smoothing
- kernel estimators are biased
- making the estimate smoother increases bias, decreases variance

- choice of kernel function, $K(\cdot)$, controls smoothness and "local-ness"
- Faraway recommends Epanechnikov kernel $K(x) = \frac{3}{4}(1 - x^2), |x| \leq 1$
- `ksmooth(base)` offers only uniform (box) or normal
- `bkde(KernSmooth)` offers `normal, box, epanech, biweight, triweight`
- `biweight`: $K(x) \propto (1 - |x|^2)^2, |x| \leq 1$ `triweight`: $K(x) \propto (1 - |x|^2)^3, |x| \leq 1$

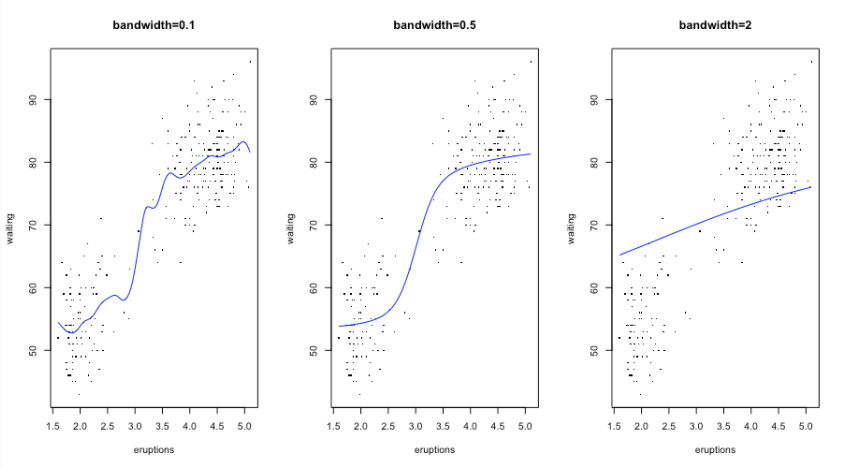
Examples



```
exb <- data.frame(exb)

plota <- ggplot(exa) + geom_point(aes(x,y)) +
geom_line(aes(x,m))+ ggtitle("Example A")

plotb <- ggplot(exb) + geom_point(aes(x,y)) +
geom_line(aes(x,m))+ ggtitle("Example B")

plotc <- ggplot(faithful) + geom_point(aes(eruptions,waiting)) +
 ggtitle("Old Faithful")

grid.arrange(plota, plotb, plotc, nrow=1) #in gridExtra library
```


Example A
y
x
1
0
-1
0.00
0.25
0.50
0.75
1.00
Example B
y
x
12
8
4
0
0.00
0.25
0.50
0.75
1.00
Old Faithful
waiting
eruptions
90
80
70
60
50
2
3
4
5

```
with(faithful, plot(eruptions, waiting, cex=2, main = "bandwidth=0.1", pch="."))
lines(locpoly(faithful$eruptions,faithful$waiting,drv=0L,
  degree=0,bandwidth=.1), col = "blue")

with(faithful, plot(eruptions, waiting, cex=2, main = "bandwidth=0.5", pch="."))
lines(locpoly(faithful$eruptions,faithful$waiting,drv=0L,
  degree=0, bandwidth=.5), col = "blue")

with(faithful, plot(eruptions, waiting, cex=2, main = "bandwidth=2", pch="."))
lines(locpoly(faithful$eruptions,faithful$waiting,drv=0L,
  degree=0, bandwidth=2), col = "blue")
```



These are smoother than the plots in ELM using `base::ksmooth`

- Nadaraya-Watson: $\hat{f}_\lambda(x) = \Sigma w_i y_i / \Sigma w_i$; $w_i = \frac{1}{\lambda} K(\frac{x_i - x_0}{\lambda})$

- $\hat{f}_\lambda(x)$ is biased

$$\mathrm{E}\{\hat{f}_\lambda(x)\} \doteq \frac{1}{2}\lambda^2 f''(x)$$

$$\mathrm{var}\{\hat{f}_\lambda(x)\} \doteq \frac{\sigma^2}{n\lambda f_\lambda(x)} \int K^2(u)du$$

- could choose λ to minimize MSE $=$ bias^2 $+$ var, at x
- could choose λ to minimize integrated MSE

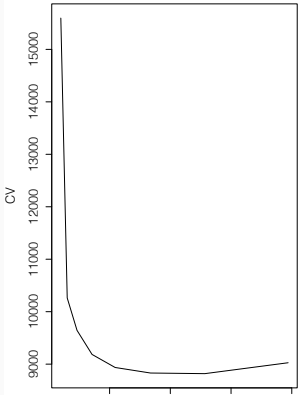
- more usual to use cross-validation

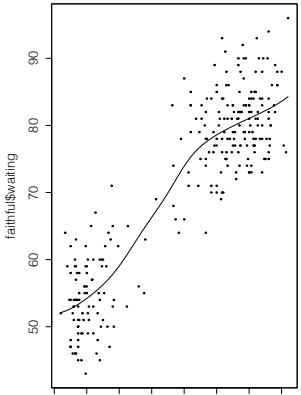
SM 10.7.1 (no n); ELM 11.1

$$CV(\lambda) = \frac{1}{n}\sum_{i=1}^{n}\{y_i - \hat{f}_{-i}(x_i)\}^2$$

Cross-validation

```
library(sm)
hm <- hcv(faithful$eruptions,
   faithful$waiting, display = "lines"
sm.regression(faithful$eruptions,
   faithful$waiting, h = hm,
   xlab = "eruptions",
   ylab = "waiting")
```





- above uses local averaging based on kernel function
- better estimates can be obtained using local regression at point x

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} 1 & (x_1 - x_0) & \cdots & (x_1 - x_0)^k \\ \vdots & \vdots & & \vdots \\ 1 & (x_n - x_0) & \cdots & (x_n - x_0)^k \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_k \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix},$$

-

$$\hat{\beta} = (X^T W X)^{-1} X^T W y, \quad W = \mathrm{diag}(w), \quad w_i = \frac{1}{\lambda} K(\frac{x_i - x_0}{\lambda})$$

-

$$\hat{f}_\lambda(x_0) = \hat{\beta}_0$$

- usually evaluate the function at sample points: $\hat{f}_\lambda(x_i), i = 1, \ldots, n$

- odd-order polynomials work better than even; usually local linear fits are used

- kernel function is often a Gaussian density, or the tricube kernel
$$K(u) = (1 - |u|^3)^3, \quad |u| \leq 1$$

- as with N-W (local averaging) estimators, choice of bandwidth controls smoothness
- **loess** is the most widely used, and is the default in `ggplot2`

- fits a local linear regression, but not by least squares

- uses a robust version of least squares that downweights outliers
- the result is that the bandwidth can change with x

- $\hat{\beta} = (X^T W X)^{-1} X^T W y \qquad W = \text{diag}(w_1, \ldots, w_n)$
- $\hat{f}_\lambda(x_0) = \hat{\beta}_0 = \sum_{i=1}^n S(x_0; x_i, \lambda) y_i$
- $S(x_0; x_1, \lambda), \ldots, S(x_0; x_n, \lambda)$ first row of "hat" matrix
- this makes it relatively easy to analyse the behaviour of local polynomial smoothers

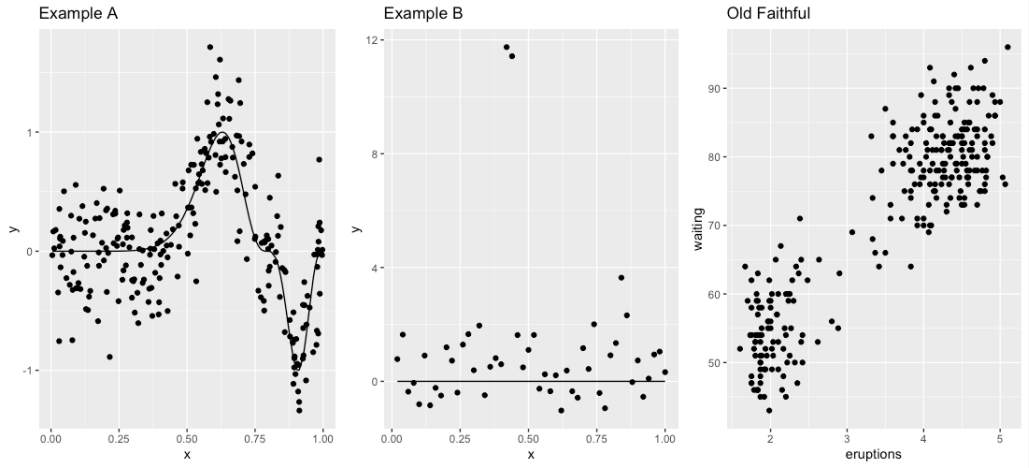
SM §10.7

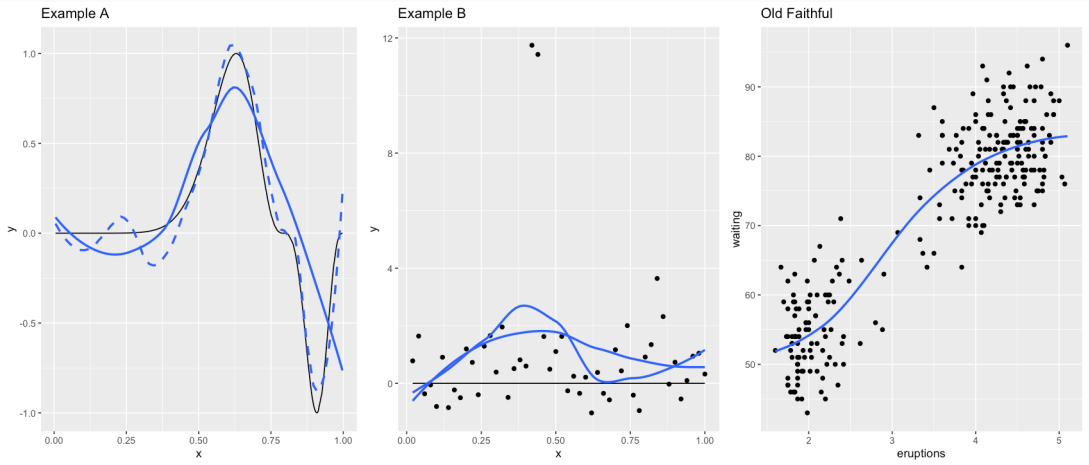
- and to simplify the expression for the cross-validation criterion $CV(\lambda)$
- fitting at each sample value gives

$$\hat{f}_\lambda(x_i) = \sum_{j=1}^n S(x_i; x_j, \lambda) y_j$$

- $$CV(\lambda) = \sum_{i=1}^{n} \{y_i - \hat{f}_{-i}(x_i)\}^2$$
- for local polynomials
$$CV(\lambda) = \sum_{i=1}^{n} \left\{ \frac{y_i - \hat{f}_\lambda(x_i)}{1 - S_{ii}(\lambda)} \right\}^2$$
- even simpler
$$GCV(\lambda) = \sum_{i=1}^{n} \left\{ \frac{y_i - \hat{f}_\lambda(x_i)}{1 - \mathrm{tr}(S_\lambda)/n} \right\}^2$$
- $$\hat{f}_\lambda(x_i) = \sum_{j=1}^{n} S(x_i; x_j, \lambda) y_j$$

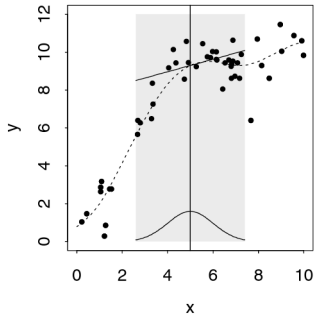
Examples


Example A
y
x
1
0
-1
0.00
0.25
0.50
0.75
1.00
Example B
y
x
12
8
4
0
0.00
0.25
0.50
0.75
1.00
Old Faithful
waiting
eruptions
90
80
70
60
50
2
3
4
5



`geom_smooth` in `ggplot` uses local polynomial fitting

robustified



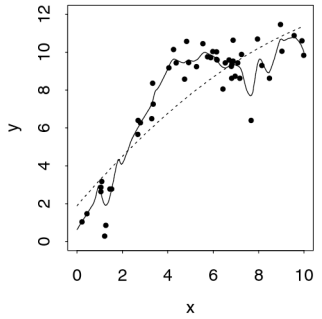


Figure 10.14 Construction of a local linear smoother. Left panel: observations in the shaded part of the panel are weighted using the kernel shown at the foot, with $h = 0.8$, and the solid straight line is fitted by weighted least squares. The local estimate is the fitted value when $x = x_0$, shown by the vertical line. Two hundred local estimates formed using equi-spaced x_0 were interpolated to give the dotted line, which is the estimate of $g(x)$. Right panel: local linear smoothers with $h = 0.2$ (solid) and $h = 5$ (dots).

Recall that a kernel function $w(u)$ is a unimodal density function symmetric about $u = 0$ and with unit variance. One choice of w is the standard normal density. Another is a rescaled form of the *tricube* function

$$w(u) = \begin{cases} (1 - |u|^3)^3, & |u| \le 1, \\ 0, & \text{otherwise}, \end{cases} \qquad (10.37)$$

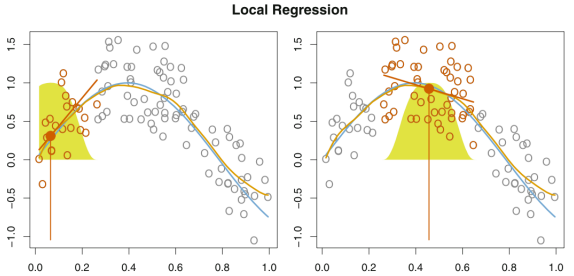


FIGURE 7.9. *Local regression illustrated on some simulated data, where the blue curve represents $f(x)$ from which the data were generated, and the light orange curve corresponds to the local regression estimate $\hat{f}(x)$. The orange colored points are local to the target point x_0, represented by the orange vertical line. The yellow bell-shape superimposed on the plot indicates weights assigned to each point, decreasing to zero with distance from the target point. The fit $\hat{f}(x_0)$ at x_0 is obtained by fitting a weighted linear regression (orange line segment), and using the fitted value at x_0 (orange solid dot) as the estimate $\hat{f}(x_0)$.*

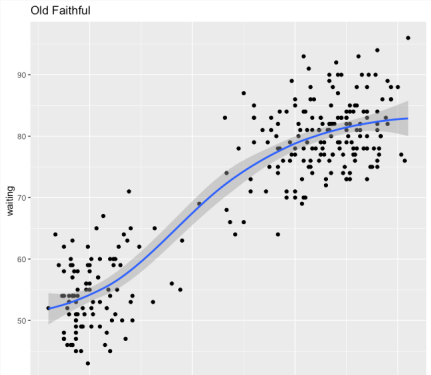
Inference after fitting local polynomials

- model: $y_i = f(x_i) + \epsilon_i, \quad i = 1, \ldots, n; \mathrm{E}(\epsilon_i) = 0; \mathrm{var}(\epsilon_i) = \sigma^2$
- $\hat{f}_\lambda(x_0) = \hat{\beta}_0 = \sum_{i=1}^n S(x_0; x_i, \lambda) y_i$
- $\mathrm{E}\{\hat{f}_\lambda(x_0)\} =$
- $\mathrm{var}\{\hat{f}_\lambda(x_0)\} =$
- how many parameters did we fit?
- by analogy with least squares, estimates of 'degrees of freedom' are $\nu_1 = \mathrm{tr}(S_\lambda)$, or $\nu_2 = \mathrm{tr}(S_\lambda^T S_\lambda)$

$$\tilde{\sigma}^2 = \frac{1}{n - 2\nu_1 + \nu_2} \sum \{y_i - \hat{f}_\lambda(x_i)\}^2$$

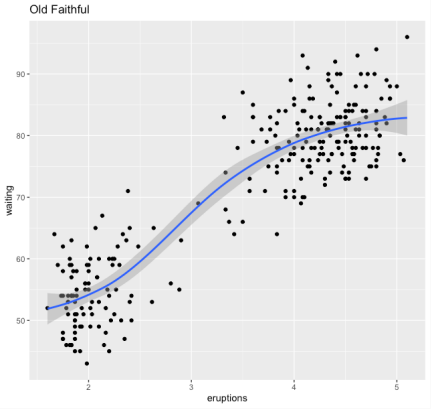
- $\mathrm{E}\{\hat{f}_\lambda(x_0)\} = \sum_{i=1}^n S(x_0; x_i, \lambda) f(x_i), \qquad \mathrm{var}\{\hat{f}_\lambda(x_0)\} = \sigma^2 \sum_{i=1}^n S^2(x_0; x_i, \lambda)$

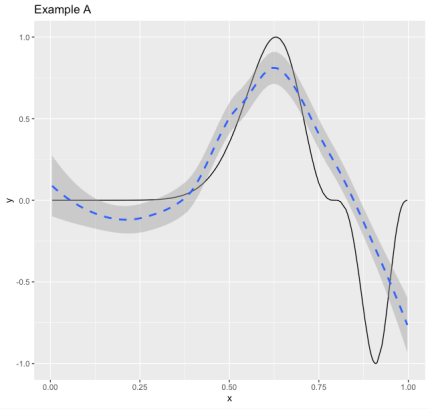
$$\frac{\hat{f}_\lambda(x_0) - \mathrm{E}\{\hat{f}_\lambda(x_0)\}}{\widehat{\mathrm{var}}\{\hat{f}_\lambda(x_0)\}^{1/2}} \dot{\sim} N(0, 1)$$



… inference after fitting local polynomials

SM §10.7





Regression splines

- model $y_i = f(x_i) + \epsilon_i$ $\quad$ $f(\cdot)$ "flexible"
- above $f(\cdot)$ is estimated at several points using local constants or local linear regression

locpoly

- another popular approach is to use some very flexible, but parametric form, for f
- for example, $f(x) = \sum_{m=1}^{M} \beta_m \phi_m(x)$
- examples of ϕ_m: $1, x, x^2, x^3$; $\quad$ $1, \sin(x), \cos(x), \sin(2x), \cos(2x)$;
- piecewise polynomials: e.g. knots at $\xi_1, \xi_2 \in [0, 1]$
- basis functions $\phi(x) : 1, x, x^2, = x^3, (x - \xi_1)^3_+, (x - \xi_2)^3_+$

- ELM p.219 builds these "by hand"
- `splines::bs()` builds cubic splines automatically

Smoothing splines

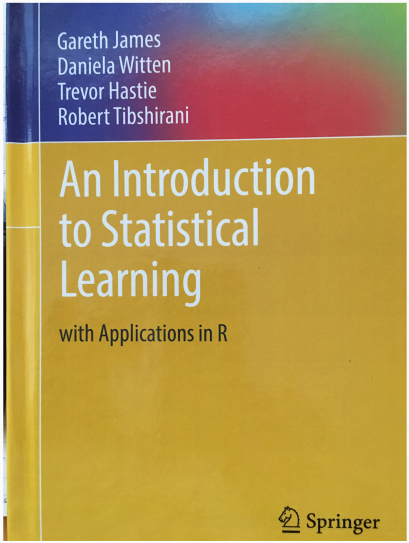
- $y_i = f(x_i) + \epsilon_i, \quad i = 1, \ldots, n$
- choose $f(\cdot)$ to solve

$$\min_f \sum_{i=1}^{n} \{y - f(x_i)\}^2 + \lambda \int_a^b \{f''(t)\}^2 dt, \quad , \lambda > 0$$

- solution is a cubic spline, with knots at each observed x_i value

see SM Figure 10.18 for a non-regularized solution

- has an explicit, finite dimensional solution
- $\hat{f} = \{\hat{f}(x_1), \ldots, \hat{f}(x_n)\} = (I + \lambda K)^{-1} y$
- K is a symmetric $n \times n$ matrix of rank $n - 2$


Gareth James
Daniela Witten
Trevor Hastie
Robert Tibshirani
An Introduction to Statistical Learning
with Applications in R
Springer

- $y_i = f(x_i) + \epsilon_i$

- local polynomial regression – `stats::loess, KernSmooth::locpoly` ELM 11.3; SM 10.7.1
- regression splines – `splines::bs , splines::ns` ELM 11.2b p 218ff
- smoothing splines – `stats:smooth.spline` ELM 11.2a; SM 10.7.2

- penalized splines – `pspline::smooth.Pspline` Peng et al. 2006
- wavelets – `wavethresh::wd` ELM 11.4
- and more... ELM 11.5; ISLR Ch.7

- same ideas can be applied to generalized linear models
- replace linear predictor $\eta_i = x_i^T \beta$ with $f(x_i)$
- use local poly, reg splines, etc. SM Ex. 10.32 logistic regression

Explanation vs Prediction

- regression (and other) models may be fit in order to uncover some structural relationship between the response and one or more predictors
  - How do wages depend on education?
  - How does numeracy score affect probability of saying yes to vaccine?
- statistical analysis will focus on estimation and/or testing
- it is a remarkable fact that the data provides both an estimate of a model parameter and an estimate of uncertainty

- the focus might instead be on predicting responses for new values of x
- or classifying new observations on the basis of their x values
- the statistical analysis will focus on the accuracy and precision of the prediction/classification
- the data used to fit the model does not provide a good assessment of the prediction or classification error — motivates the division of data into training and test sets