Chapter 11

Adapted from Timothy Hanson

Department of Statistics, University of South Carolina

Stat 704: Data Analysis I

11.1: Weighted least squares

- Chapters 3 and 6 discuss transformations of $x_1, \dots, x_k$ and/or Y.
- This is "quick and dirty" but may not solve the problem.
- Or can create an uninterpretable mess (book: "inappropriate").
- More advanced remedy: *weighted least squares* regression.
- Model is as before

$$Y_i = \beta_0 + \beta_1 x_{i1} + \cdots \beta_k x_{ik} + \epsilon_i,$$

but now

$$\epsilon_i \overset{ind.}{\sim} N(0, \sigma_i^2).$$

Note the subscript on σ_i...

- Here $\text{var}(Y_i) = \sigma_i^2$. Give observations with higher variance *less weight* in the regression fitting.
- Say $\sigma_1, \dots, \sigma_n$ are known. Let $w_i = 1/\sigma_i^2$ and define the weight matrix

$$\mathbf{W} = \begin{bmatrix} w_1 & 0 & \cdots & 0 \\ 0 & w_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_n \end{bmatrix} = \begin{bmatrix} \sigma_1^{-2} & 0 & \cdots & 0 \\ 0 & \sigma_2^{-2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_n^{-2} \end{bmatrix}.$$

- Maximizing the likelihood (pp. 422-423) gives the estimates for $\boldsymbol{\beta}$:

$$\mathbf{b}_w = (\mathbf{XWX}')^{-1}\mathbf{X}'\mathbf{WY}.$$

- However, $\sigma_1, \ldots, \sigma_n$ are almost always unknown.
- If the mean function is appropriate, then $E(e_i^2) = \sigma_i^2(1 - h_{ii})$ where e_i is obtained from ordinary least squares, so e_i^2 estimates σ_i^2 and $|e_i|$ estimates σ_i (pp. 424-425) as $h_{ii} \to 0$ as $n \to \infty$.
- Look at plots of $|e_i|$ from a normal fit against predictors in the model and the fitted values $\hat{Y}_i$ to see how σ_i changes with predictors or fitted values.
- For example, if $|e_i|$ increases linearly with $\hat{Y}_i = \mathbf{x}_i'\mathbf{b}$, then we'll fit $|e_i| = \alpha_0 + \alpha_1 x_{i1} + \cdots + \alpha_k x_{ik} + \delta_i$ and obtain the fitted values $\widehat{|e_i|}$.
- If e_i^2 increases linearly with only x_{i4}, then we'll fit $e_i^2 = \alpha_0 + \alpha_4 x_{i4} + \delta_i$ and obtain the fitted values $\widehat{e_i^2}$.

1. Regress Y against predictor variable(s) as usual (OLS) & obtain $e_1, \ldots, e_n$ & $\hat{Y}_1, \ldots, \hat{Y}_n$.
2. Regress $|e_i|$ against predictors $x_1, \ldots, x_k$ or fitted values $\hat{Y}_i$.
3. Let $\hat{s}_i$ be the fitted values for the regression in 2.
4. Define $w_i = 1/\hat{s}_i^2$ for $i = 1, \ldots, n$.
5. Use $\mathbf{b}_w = (\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}\mathbf{X}'\mathbf{W}\mathbf{Y}$ as estimated coefficients – automatic in SAS. SAS will also use the correct $\text{cov}(\mathbf{b}_w) = (\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}$ (p. 423). This is developed formally in linear models.

SAS code: initial fit

```
* SAS example for Weighted Least Squares ;
* Blood pressure data in Table 11.1      ;
data bloodp; input age dbp @@; datalines;
  27   73  21   66  22   63  24   75  25   71  23   70  20   65
  20   70  29   79  24   72  25   68  28   67  26   79  38   91
  32   76  33   69  31   66  34   73  37   78  38   87  33   76
  35   79  30   73  31   80  37   68  39   75  46   89  49  101
  40   70  42   72  43   80  46   83  43   75  44   71  46   80
  47   96  45   92  49   80  48   70  40   90  42   85  55   76
  54   71  57   99  52   86  53   79  56   92  52   85  50   71
  59   90  50   91  52  100  58   80  57  109
; run;

* Regress the response, dbp, against the predictor, age ;
* The plots show definite nonconstant error variance    ;
proc reg data=bloodp;
 model dbp=age;
 output out=temp r=residual;
run;
```

SAS code: determining w_i

```
* Plot of absolute residuals against age shows that
  absolute residuals increase approximately linearly;
data temp; set temp; absr = abs(residual); run;
proc sgplot data=temp;
scatter x=age y=absr/markerattrs=(color=blue symbol=Diamond);
loess x=age y=absr/nomarkers lineattrs=(color=blue);
xaxis label="Age";
yaxis label="Absolute Residuals";
```

SAS code: WLS fit

```
* Regress absolute residuals against the age           ;
* This second regression is done on the data set temp  ;
proc reg data=temp;
  model absr=age;
  output out=temp1 p=s_hat ;
run;

* Define weights using the fitted values from this second regression ;
data temp1; set temp1; w=1/(s_hat**2); run;

* Using the WEIGHT option in PROC REG to get the WLS estimates ;
* This last regression is done on the data set temp1           ;
proc reg data=temp1;
  weight w;
  model dbp=age / clb;
  output out=temp2 r=residual;
run;
```

- se(b_1) reduced from 0.097 (OLS) to 0.079 (WLS). WLS verified via bootstrap on pp. 462–463 (just FYI).
- R^2 no longer interpreted the same way in terms of amount of total variability explained by model.
- In WLS, standard inferences about coefficients may not be valid for small sample sizes when weights are estimated from the data.
- If MSE of the WLS regression is near 1, then our estimation of the "error standard deviation" function is okay. Here it's 1.21.

Fitting the model directly...

- A drawback of this approach is that the weights $w_i = 1/\hat{s}_i^2$ have associated variability that is not reflected in $\text{cov}(\mathbf{b}_w)$.
- It is possible to fit the implied model

  $$Y_i = \beta_0 + \beta_1 a_i + \epsilon_i, \quad \epsilon_i \sim N(0, \tau_0 + \tau_1 a_i),$$

  directly in SAS. One option is to have SAS maximize the associated likelihood in PROC NLMIXED.
- Note that a similar, and possibly more appropriate, model

  $$Y_i = \beta_0 + \beta_1 a_i + \epsilon_i, \quad \epsilon_i \sim N(0, e^{\tau_0 + \tau_1 a_i}),$$

  was used for the Breusch-Pagan test $H_0 : \tau_1 = 0$ described in Sections 3.6 and 6.8. This model can also be fit easily in PROC NLMIXED.
- However, things like F-tests go out the window and everything relies on asymptotics (which for large enough samples work fine).

SAS code: fitting model directly

```
* Model fit directly using PROC NLMIXED             ;
* Starting values obtained from regressions 1 and 2 ;
proc nlmixed data=bloodp;
 parms beta0=50 beta1=0.5 tau0=-1 tau1=0.2;
 mu=beta0+beta1*age; sigma=tau0+tau1*age;
 model dbp ~ normal(mu,sigma*sigma);
run;
```

11.2: Ridge regression

- Before considering ridge regression, recall that even *serious multicollinearity* does not present a problem when the focus is on prediction, and prediction is limited to the overall pattern of predictors in the data. Use $\mathbf{x}_h'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{x}_h$ for predictor $\mathbf{x}_h$ and compare to the rest of the leverages.
- Principle components provide composite "predictors" that are uncorrelated. Under umbrella term of "dimension reduction."
- **Ridge regression** is an advanced remedial measure for multicollinearity that uses a *biased* estimate $\mathbf{b}^R$ instead of the OLS $\mathbf{b}$.
- Although biased, it may have *less variance* – one of the effects of multicollinearity was exploding se(b_k). See Fig. 11.2 (p. 432).

- Ridge regression adds a biasing constant c to the normal equations based on the standardized regression model developed in Section 7.5 (also used for VIFs in 10.5); read pp. 273–275 and p. 433.
- $c = 0 \Rightarrow$ OLS estimator $\mathbf{b}$.
- Bias in the estimator $\mathbf{b}^R$ increases/decreases with c.
- VIFs/R^2 decrease with increasing c.
- Look at plots of b_j^R and VIF_j versus c to see when estimates and variance inflation stabilize. Can get these automatically in SAS.
- Note no standard errors when choosing c by eye. Need to use bootstrap; not automatic in SAS.
- Ridge regression is related to the LASSO; more in a minute...

Standard error for fixed c

Page 433. Working with standardized model

$$Y_i^* = \beta_1^* x_{i1}^* + \cdots \beta_k^* x_{ik}^* + \epsilon_i^*.$$

$$\mathbf{b}^R = ((\mathbf{X}^*)'\mathbf{X}^* + c\mathbf{I})^{-1}(\mathbf{X}^*)'\mathbf{Y}^*.$$

So

$$cov(\mathbf{b}^R) = ((\mathbf{X}^*)'\mathbf{X}^* + c\mathbf{I})_k^{-1}(\mathbf{X}^*)'(\mathbf{X}^*)((\mathbf{X}^*)'\mathbf{X}^* + c\mathbf{I})^{-1}(\sigma^*)^2.$$

Why not output from SAS?
Note: Ridge regression gives the same estimate as the Bayesian posterior mode of $\boldsymbol{\beta}^*$ under independent mean-zero normal priors with variance τ^2 on the $\beta_1^*, \ldots, \beta_k^*$. Here, $c = (\sigma^*)^2/\tau^2$.

SAS code & output: body fat data

```
*******************************
*  Body fat data from Chapter 7
*******************************;
data body;
  input triceps thigh midarm bodyfat @@;
  cards;
  19.5  43.1  29.1  11.9  24.7  49.8  28.2  22.8
  30.7  51.9  37.0  18.7  29.8  54.3  31.1  20.1
  19.1  42.2  30.9  12.9  25.6  53.9  23.7  21.7
  31.4  58.5  27.6  27.1  27.9  52.1  30.6  25.4
  22.1  49.9  23.2  21.3  25.5  53.5  24.8  19.3
  31.1  56.6  30.0  25.4  30.4  56.7  28.3  27.2
  18.7  46.5  23.0  11.7  19.7  44.2  28.6  17.8
  14.6  42.7  21.3  12.8  29.5  54.4  30.1  23.9
  27.7  55.3  25.7  22.6  30.2  58.6  24.6  25.4
  22.7  48.2  27.1  14.8  25.2  51.0  27.5  21.1
;
run;

proc reg data=body outest=ridge outvif ridge=0.01 to 0.5 by .01;
 model bodyfat=triceps thigh midarm;
 plot / ridgeplot; run;
* I would probably take c=0.1 or c=0.2 based on the plot;
proc print; run;
proc reg data=body outest=ridge ridge=0.2;
 model bodyfat=triceps thigh midarm; run;
proc print data=ridge; run;
```

Ridge regression in R

`lm.ridge` provides a function for performing ridge regression in R. You can use generalized cross-validation (Golub, Heath, and Wahba, 1979 *Technometrics*) to choose the best c. This is preferable to PRESS. A newer package `ridge` uses a different method for choosing c and provides p-values for the best ridge model.

```
library(MASS)
bodyfat=read.table("http://www.stat.sc.edu/~hansont/stat704/bodyfat.txt",
 header=T)
f=lm.ridge(bodyfat~triceps+thigh+midarm,data=bodyfat,lambda=seq(0,2,by=0.005))
plot(f)
select(f) # gives c=0.02
f=lm.ridge(bodyfat~triceps+thigh+midarm,data=bodyfat,lambda=0.02)
coef(f) # no standard errors...BOOOO!!!

library(ridge) # uses c selection based on PCA
f=linearRidge(bodyfat~triceps+thigh+midarm,data=bodyfat)
summary(f) # p-values!!!  hooray!!!
```

Penalized least-squares (p. 436) formulation of ridge regression:

$$Q_{pen} = \sum_{i=1}^{n} (Y_i^* - (\mathbf{x}_i^*)'\mathbf{b}^R)^2 + c \sum_{j=1}^{k} (b_j^R)^2.$$

The solution is $\mathbf{b}^R$ that minimizes Q_{pen}.

LASSO chooses $\mathbf{b}^L$ to minimize

$$\sum_{i=1}^{n}(Y_i^* - (\mathbf{x}_i^*)'\mathbf{b}^L)^2 + c\sum_{j=1}^{k}|b_j^L|$$

In LASSO, this constraint leads to some b_j^L's set exactly to zero, so LASSO can be viewed as a method of variable selection as well as coefficient estimation.

Traditionally ridge regression estimates have been easier to obtain (computationally) than LASSO estimates. However, recent advances allow for the routine use of LASSO. LASSO for variable selection is in the new SAS PROC GLMSELECT.

LASSO on the bodyfat data

```
proc glmselect data=body plot=coefficients;
* can also have class statement;
* default for LASSO picks model w/ smallest BIC (i.e. SBC);
* plot is each coefficient as c is increased;
 model bodyfat=triceps thigh midarm / selection=lasso;
run;
```

PROC GLMSELECT stops with the model that has the lowest BIC.

Compare the LASSO coordinate evolution plot to that obtained via ridge regression. Question: are the coefficients for the standardized model, or unstandardized? Looks like the latter.

In R packages the biasing constant (and therefore $\mathbf{b}^L$) can be estimated via cross-validation, but not in SAS.