STAT 420: Methods of Applied Statistics

Model Diagnostics — Multicollinearity

Ruoqing Zhu, Ph.D. <rqzhu@illinois.edu>

Course website: https://sites.google.com/site/teazrq/teaching/STAT420

Department of Statistics
University of Illinois at Urbana-Champaign
April 4, 2017

Problems with the design matrix

- We discussed diagnostics on the error terms and the linear functional form.
- In many real applications, “bad” design matrix will also cause trouble.
- Recall that our assumptions on the design matrix $\mathbf{X}$ is: fixed value and full rank.
- What if $\mathbf{X}$ is not full rank, or “very close” to singular?

Exact Collinearity

- When the covariates are exactly linearly dependent, we run into model identification problem.
- Suppose $X_3 = aX_1 + bX_2$, then the linear model

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3$$

  could simply be reformulated into

$$\begin{aligned} Y &= \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 \big(aX_1 + bX_2\big) + \epsilon \\ &= \beta_0 + (\beta_1 + a\beta_3) X_1 + (\beta_2 + b\beta_3) X_2 + \epsilon, \end{aligned}$$

  which makes the regression identical to the one that using just X_1 and X_2.
- In $\mathrm{lm()}$ this type of exactly linearly dependent is detected automatically. See our R code.

Collinearity

- In practice, we often see highly correlated predictors, rather than exactly linearly dependent ones.
- This may cause even more trouble
- What do you expect to get from the following model fitting?

```
> set.seed(1)
> x1 = rnorm(n)
> x2 = rnorm(n) # x1 and x2 are independent
> x3 = 1 + 2 * x1 + 3 * x2 + rnorm(n, sd = 0.01)
> y = 3 + x1 + x2 + x3 + rnorm(n)
>
> mydata = data.frame(x1, x2, x3, y)
>
> fit = lm(y~., data = mydata)
> summary(fit)
```

Collinearity

```
> summary(fit)

Call:
lm(formula = y ~ ., data = mydata)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    6.999      7.276   0.962    0.337
x1             9.290     14.563   0.638    0.524
x2            13.304     21.834   0.609    0.543
x3            -3.103      7.279  -0.426    0.670

Residual standard error: 1.096 on 196 degrees of freedom
Multiple R-squared:  0.953, Adjusted R-squared:  0.9523
F-statistic:  1326 on 3 and 196 DF,  p-value: < 2.2e-16
```

Collinearity

- The model has $R^2 = 0.953$, which indicates a very good fitting.
- However, non of the variables are significant.
- Recall that the estimated variance of $\widehat{\boldsymbol{\beta}}$ is $\widehat{\sigma}^2(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}$
- $\widehat{\sigma}^2$ is around 1, hence $(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}$ is very large. (why?)
- Lets investigate the matrix $\mathbf{X}^\mathsf{T}\mathbf{X}$.

Collinearity

- When $\mathbf{A}$ is a symmetric matrix, we have

$$\mathbf{A} = \mathbf{Q}\mathbf{D}\mathbf{Q}^\mathsf{T},$$

  where $\mathbf{D}$ is a diagonal matrix with the eigenvalues and

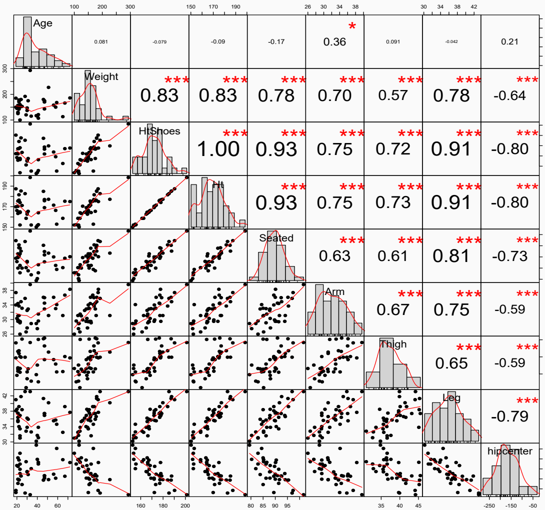
$$\mathbf{A}^{-1} = \mathbf{Q}\mathbf{D}^{-1}\mathbf{Q}^\mathsf{T},$$

  Hence, if $\mathbf{D}$ has very small positive values, $\mathbf{A}^{-1}$ will be very large.
- This is the case for $\mathbf{X}^\mathsf{T}\mathbf{X}$ and $(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}$.
- Since $\mathbf{X}^\mathsf{T}\mathbf{X}$ is "almost" singular (with some very small eigenvalues), $(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}$ is very large, making the variance of $\widehat{\beta}$ large.

Example: seatpos

- The seatpos data contains useful information for car manufacturers considering comfort and safety when designing a car seat.
- The predictors in this dataset are various attributes of car drivers, such as their height, weight and age.
- The response variable hipcenter measures the "horizontal distance of the midpoint of the hips from a fixed location in the car in mm." — the position of the seat.
- Lets first investigate the correlations

A picture


Age
Weight
HtShoes
Ht
Seated
Arm
Thigh
Leg
hipcenter
0.081
-0.079
-0.09
-0.17
0.36
0.091
-0.042
0.21
0.83
0.83
0.78
0.70
0.57
0.78
-0.64
1.00
0.93
0.75
0.72
0.91
-0.80
0.93
0.75
0.73
0.91
-0.80
0.63
0.61
0.81
-0.73
0.67
0.75
-0.59
0.65
-0.59
-0.79

Example: seatpos

- From our previous intuition, two variables are worth investigating: HtShoes and Ht — height with shoes and height, which should be almost perfectly correlated. The estimated correlation is 1 (after rounding).
- This should again lead to very small eigenvalues in $\mathbf{X}^\mathsf{T}\mathbf{X}$, and can be easily verified.
- The model fitting results show highly significant F value (see R code), however, none of the predictors are significant. This is suspicious.

Variance Inflation Factor

- How to detect these problem and select a good model?
- We turn to observe a fact that the variance of $\widehat{\beta}_j$ can be written as

$$\mathrm{Var}(\widehat{\beta}_j) = \sigma^2 \left(\frac{1}{1 - R_j^2} \right) \frac{1}{(n-1)s_j^2}$$

  where s_j^2 is the variance of X_j and R_j^2 is the proportion of variation in the jth predictor explained by the other predictors.
- Essentially, R_j^2 is the R^2 of the regression of X_j on all other predictors.
- This is due to a fact of the block matrix inverse, which is in the lecture note Intro.

Variance Inflation Factor

- As we can see, if a variable can be mostly explained by other predictors, then R_j^2 is close to 1. Hence the variance of a beta estimation is greatly inflated since $\frac{1}{1-R_j^2}$ is large.
- The variance inflation factors (VIF) measures the extent to which the variance is inflated due to predictor correlations:

$$\text{VIF} = \frac{1}{1 - R_j^2}$$

- In practice, VIF > 5 are considered problematic.
- In the seatpos data, both HtShoes and Ht have VIF > 300. What should we do?
- Keep in mind that removing any one variable will change the VIF of all others.

Ridge Regression

- The idea of Ridge is to force the matrix $\mathbf{X}^\mathsf{T}\mathbf{X}$ away from singular, by adding a diagonal matrix $\lambda\mathbf{I}$
- Then, our solution of the ridge regression is simply

$$\widehat{\beta}^{\text{ridge}} = (\mathbf{X}^\mathsf{T}\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\mathsf{T}\mathbf{y}$$

- This is called "shrinkage" method. However, it introduces bias into the regression estimates.
- This can be done using the lm.ridge() function in the MASS package. We will inevitably choose a tuning parameter λ.

Ridge regression and tuning parameter

```
> # be careful that the ridge regression will first scale the
    predictors to sd = 1,
> # and then apply the ridge techique
>
> ridge.fit = lm.ridge(hipcenter~., data = seatpos, lambda = seq
    (1, 100, 1))
> plot(ridge.fit)
> # this helps to select the best tuning parameter
> which.min(ridge.fit$GCV)
22
22
> lm.ridge(hipcenter~., data = seatpos, lambda= 22)
```