Multicollinearity

San-Teng Huang,Shang-Chien Ho,Hsing-Cheng Pan

National Dong Hwa University

2018/12/19

Outline

Multiple Linear Regression

- Consider multiple linear model:

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \ldots + \beta_{p-1} X_{ip-1} + \varepsilon_i \quad , i = 1, 2, \ldots, n \tag{1}$$

where $\boldsymbol{\beta}$ is $p \times 1$ vector, $\mathbf{X}$ is $n \times p$ matrix,
and ε_i *i.i.d* with mean 0, variance σ^2, for $i = 1, \ldots, n$.

Multiple Linear Regression

- The coefficients of the estimates:

$$\hat{\beta} = \left(X^T X\right)^{-1} X^T y$$

- The variance of the estimates:

$$Var\left[\hat{\beta}\right] = \sigma^2 \left(X^T X\right)^{-1}$$

Even if $X^T X$ isn't singular, but is close to being non-invertible, the variances will become huge.

Collinearity

There are several equivalent conditions for X^TX,
to be singular or non-invertible:

- $\det(X^TX) = 0$.
- At least one eigenvalue of X^TX is 0.
- X^TX is rank deficient, meaning that one or more of its columns (or rows) is equal to a linear combination of the other rows.

- Geometric Perspective
- Why Multicollinearity Is Harder
- Dealing with Collinearity by Deleting Variables
- Diagnosing Collinearity Among Pairs of Variables

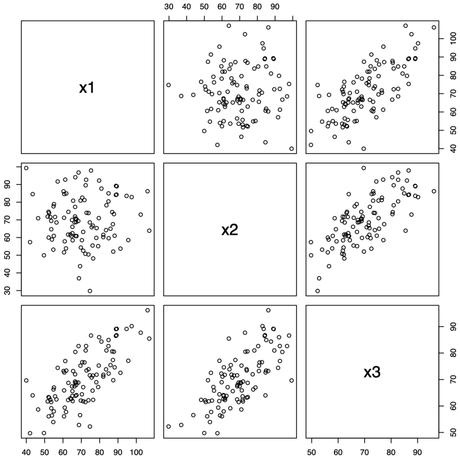


Figure: suppose X_1 and X_2 are independent Gaussians, of equal variance σ^2, and X_3 is their average,$X_3 = (X_1 + X_2)\,/2$

Multicollinearity

Multicollinearity means, $\exists \mathbf{a} \neq 0$ $s.t.$

$$a_1X_1 + a_2X_2 + \ldots + a_pX_p = \sum_{i=1}^{p} a_iX_i = a_0$$

where $\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix}$ is $p \times 1$ vector, a_0 is a constant.

That is $\mathbf{a}^T X = a_0$, for $\mathbf{a} \neq 0$.

$$Var\left[\mathbf{a}^T X\right] = 0, \quad \mathbf{a} \neq 0$$

$$\begin{aligned} Var\left[\mathbf{a}^T X\right] &= Var\left[\sum_{i=1}^{p} a_i X_i\right] \\ &= \sum_{i=1}^{p}\sum_{j=1}^{p} a_i a_j Cov\left[X_i, X_j\right] \\ &= \mathbf{a}^T Var\left[X\right] \mathbf{a} \end{aligned}$$

- The eigenvectors of Var[X], such that

$$Var\left[X\right]v_i = \lambda v_i$$

- The eigenvalues are all ≥ 0.
- Any vector can be re-written as a sum of eigenvectors:

$$\mathbf{a} = \sum_{i=1}^{p} \left(\mathbf{a}^T v_i\right) v_i$$

- The eigenvectors can be chosen so that they all have length 1 and are orthogonal to each other.
  $(||v_i|| = 1,\ and\ v_i^T v_j = 0\ for\, i \neq j)$

$$
\begin{aligned}
Var[X]\,\mathbf{a} &= Var[X]\sum_{i=1}^{p}\left(\mathbf{a}^T v_i\right) v_i \\
&= \sum_{i=1}^{p}\left(\mathbf{a}^T v_i\right) Var[X]\, v_i \\
&= \sum_{i=1}^{p}\left(\mathbf{a}^T v_i\right) \lambda_i v_i \\
\mathbf{a}^T Var[X]\,\mathbf{a} &= \left(\sum_{i=1}^{p}\left(\mathbf{a}^T v_i\right) v_j\right)^T \sum_{i=1}^{p}\left(\mathbf{a}^T v_i\right) \lambda_i v_i \\
&= \sum_{i=1}^{p}\sum_{j=1}^{p}\left(\mathbf{a}^T v_i\right)\left(\mathbf{a}^T v_i\right) v_j^T v_i \lambda_i \\
&= \sum_{i=1}^{p}\left(\mathbf{a}^T v_i\right)^2 \lambda_i = 0
\end{aligned}
$$

- The predictors are multi-collinear if and only if Var[X] has zero eigenvalues.

- Every multi-collinear combination of the predictors is either an eigenvector of Var[X] with zero eigenvalue, or a linear combination of such eigenvectors.

Finding the Eigendecomposition

- eigen(A) function in R.
- numpy.linalg.eig(A) function in Python.
- include<Eigen/Eigenvalues> in C++.
- eig(A) in matlab.

Using the Eigendecomposition

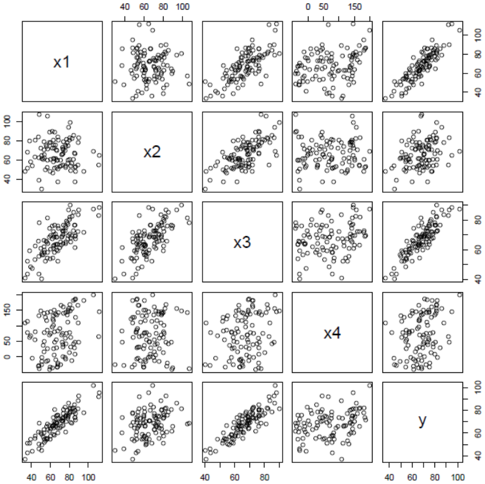
1. Find the eigenvalues and eigenvectors.
2. If any eigenvalues are zero, the data is multicollinear. If any are very close to zero, the data is nearly multicollinear.
3. Examine the corresponding eigenvectors. These indicate the linear combinations of predictors which equal constants.

Example

First make up some data which displays exact multi-collinearity. Let's say that X_1 and X_2 are both Gaussian with mean 70 and standard deviation 15, and are uncorrelated, that $X_3 = \frac{(X_1+X_2)}{2}$, and that $Y = 0.7X_1 + 0.3X_2 + \epsilon$, with $\epsilon \sim N(0, 15)$.

Second input coding below:

```
x1 <- rnorm(100, mean=70, sd=15)
x2 <- rnorm(100, mean=70, sd=15)
x3 <- (x1+x2)/2
x4 <- x1+runif(100, min=-100, max=100)
y <- 0.7*x1 + 0.3*x2 + rnorm(100, mean=0, sd=sqrt(15))
df <- data.frame(x1=x1, x2=x2, x3=x3, x4=x4, y=y)
pairs(df)
cor(df)
```


x1
x2
x3
x4
y
40 60 80 100
0 50 150
40 60 80

```
##              x1          x2        x3         x4         y
## x1  1.00000000 -0.01979669 0.7290418 0.29354541 0.8810356
## x2 -0.01979669  1.00000000 0.6699024 0.03450894 0.3263256
## x3  0.72904178  0.66990244 1.0000000 0.24161019 0.8776559
## x4  0.29354541  0.03450894 0.2416102 1.00000000 0.3006694
## y   0.88103556  0.32632563 0.8776559 0.30066941 1.0000000
```

Figure: Pairs plot and correlation matrix for the example. Notice that neither the pairs plot nor the correlation matrix reveals a problem, which is because it only arises when considering X_1,X_2,X_3 at once.

```
# Create the variance matrix of the predictor variables
var.x <- var(df[,c("x1","x2","x3","x4")])
# Find the eigenvalues and eigenvectors
var.x.eigen <- eigen(var.x)
# Which eigenvalues are (nearly) 0?
(zero.eigenvals <- which(var.x.eigen$values < 1e-12))

## [1] 4

# Display the corresponding vectors
(zero.eigenvectors <- var.x.eigen$vectors[,zero.eigenvals])

## [1]  4.082483e-01  4.082483e-01 -8.164966e-01  3.330669e-16
```

Figure: Example of using the eigenvectors of Var[X] to find collinear combinations of the predictor variables. Here, what this suggests is that $-X_1 - X_2 + 2X_3 =$ constant. This is correct, since $X3 = \frac{(X_1+X_2)}{2}$, but the eigen-decomposition didn't know this; it discovered it.

Principal Components Regression

Define new variable:

$$W_1 = v_1^T X$$
$$W_i = v_i^T X$$
$$W_p = v_p^T X$$

Where W_1 is the projection of the original data vector X onto the leading eigenvector, or the **principal component**. W_2 is the projection on the second principal component and uncorrelated with W_1. In fact, $\widehat{Cov}[W_i, W_j] = 0$ if $i \neq j$.

In **principal components regression**, we pick some $k \leq p$

$$Y_i = \gamma_{i0} + \gamma_{i1} W_{i1} + ... + \gamma_{ik} W_{ik} + \epsilon_i \qquad , i = 1, 2, 3, ..., n$$

where ϵ_i has expectation 0, constant variance, and no correlation from one observation to another.

There are a number of things to be said about principal components regression.

1. We need some way to pick k.
2. The PC regression can be hard to interpret.

Ridge Regression

- The ordinary least squares(OLS) is to

$$\min_{\beta}(\boldsymbol{Y}-\boldsymbol{X}\beta)^T(\boldsymbol{Y}-\boldsymbol{X}\beta)$$

- The ridge regression is to

$$\min_{\beta}(\boldsymbol{Y}-\boldsymbol{X}\beta)^T(\boldsymbol{Y}-\boldsymbol{X}\beta)\ with\ ||\beta||_2 \leq c,\ c > 0$$

which is equivalent to

$$\min_{\beta}(\boldsymbol{Y}-\boldsymbol{X}\beta)^T(\boldsymbol{Y}-\boldsymbol{X}\beta) + \lambda||\beta||_2,\ \lambda > 0$$

where $||\beta||_2 = \sqrt{\sum_{j=1}^{p} \beta_j^2}$.

Geometric Interpretation of Ridge Regression

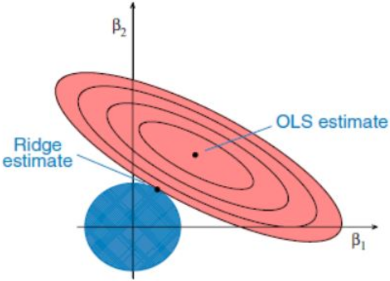


Figure: For $p = 2$, the ellipses correspond to the contours of residual sum of squares (SSE), and SSE is minimized at ordinary least square (OLS) estimate.

- The ridge regression estimator is

$$\hat{\beta}_\lambda = (\boldsymbol{X}^T\boldsymbol{X} + \lambda \boldsymbol{I})^{-1}\boldsymbol{X}^T\boldsymbol{Y}$$

where $\lambda > 0$ is a tuning parameter.

- Note:
  1. If $\lambda = 0$, then $\hat{\beta}_\lambda = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Y}$
  2. If $\lambda \to \infty$, then $\hat{\beta}_\lambda \to 0$,

  i.e. the larger λ, the smaller β_j's value you will get.
- This would break any exact multicollinearity, so the inverse always exists.

- The ridge regression estimator $\hat{\beta}_\lambda$ is biased.

$$
\begin{aligned}
E(\hat{\beta}_\lambda) &= (\boldsymbol{X}^T\boldsymbol{X} + \lambda \boldsymbol{I})^{-1}\boldsymbol{X}^T E(\boldsymbol{Y}) \\
&= (\boldsymbol{X}^T\boldsymbol{X} + \lambda \boldsymbol{I})^{-1}\boldsymbol{X}^T \boldsymbol{X}\beta \\
Var[\hat{\beta}_\lambda] &= Var[(\boldsymbol{X}^T\boldsymbol{X} + \lambda \boldsymbol{I})^{-1}\boldsymbol{X}^T \boldsymbol{Y}] \\
&= Var[(\boldsymbol{X}^T\boldsymbol{X} + \lambda \boldsymbol{I})^{-1}\boldsymbol{X}^T \varepsilon] \\
&= \sigma^2(\boldsymbol{X}^T\boldsymbol{X} + \lambda \boldsymbol{I})^{-1}\boldsymbol{X}^T\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X} + \lambda \boldsymbol{I})^{-1}
\end{aligned}
$$

- How to choose the parameter λ ?
  by using cross-validation.

Problem of High-Dimensional Regression

- In high-dimensional data($n < p$), it will always have multicollinearity.
  (since $rank(\boldsymbol{X}) = n < p$ i.e. the column space of $\boldsymbol{X}$ is n, but the number of predictors is p.)
- We may
  (i)reduce the dimension until $< n$.(as in principle components regression)
  (ii)penalize the estimates to make them stable and regular.(as in ridge regression).

Conclusion

- What is multicollinearity ?
- How to deal with multicollinearity ?

  -Pairs Plot of the predictors

  -Eigendecomposition

  -Principal Components Regression

  -Ridge Regression