Big data analysis with linear models

Feng Li

feng.li@cufe.edu.cn

School of Statistics and Mathematics
Central University of Finance and Economics

May 5, 2014

Revision: May 5, 2014

Today we are going to learn...

QR decomposition

- QR decomposition is a decomposition of a matrix A into a product $A = QR$ of an orthogonal matrix Q and an upper triangular matrix R.
- Compared to the direct matrix inverse, inverse solutions using QR decomposition are more numerically stable as evidenced by their reduced condition numbers.
- To solve the underdetermined (m ¡ n) linear problem $Ax = b$ where the matrix A has dimensions $m \times n$ and rank m
  - first find the QR factorization of the transpose of A: $A^T = QR$, where Q is an orthogonal matrix (i.e. $Q^T = Q^{-1}$), and R has a special form: $R = \begin{bmatrix} R_1 \\ 0 \end{bmatrix}$. Here R_1 is a square $m \times m$ right triangular matrix, and the zero matrix has dimension $(n - m) \times m$.
  - it can be shown that a solution to the inverse problem can be expressed as: $x = Q \begin{bmatrix} (R_1^T)^{-1} b \\ 0 \end{bmatrix}$

The linear model with `biglm`

- For a linear model $y = X\hat{\beta}$, the QR approach is slightly slower than $(X'X)^{-1}Xy$ but more accurate.
- The procedure is to compute the **incremental QR decomposition** of X to get R and $Q'Y$, solve $R\beta = Q'y$.
- The variance is computed via an incremental **Huber/White sandwich variance estimator**.

The biglm implementation in R

```
biglm> data(trees)
biglm> ff<-log(Volume)~log(Girth)+log(Height)
biglm> chunk1<-trees[1:10,]
biglm> chunk2<-trees[11:20,]
biglm> chunk3<-trees[21:31,]
biglm> a <- biglm(ff,chunk1)
biglm> a <- update(a,chunk2)
biglm> a <- update(a,chunk3)

biglm> summary(a)
Large data regression model: biglm(ff, chunk1)
Sample size =  31
                Coef    (95%     CI)     SE p
(Intercept) -6.6316 -8.2312 -5.0320 0.7998 0
log(Girth)   1.9826  1.8326  2.1327 0.0750 0
log(Height)  1.1171  0.7082  1.5260 0.2044 0

biglm> deviance(a)
[1] 0.1854634
```