Review of mehtods for solving systems of linear equations

G. de los Campos

9/28/2022

Solving Systems of Linear Equations

In this handout we review various ways of solving systems of linear equations. '

We illustrate the methods in the context of linear regression via OLS; however, the methods can be used to solve other systems of equations.

1) Setting

Consider a linear regression model of the form

$\mathbf{y}_{n\times 1} = \mathbf{X}_{n\times p}\beta_{p\times 1} + \varepsilon_{n\times 1}$

The Ordinary Least Square estimates is given by the solution ($\hat{\beta}$) to the following system of equations

$\mathbf{X}'\mathbf{X}\hat{\beta} = \mathbf{X}'\mathbf{y}$ [1]

or, using a more general notation

$\mathbf{C}\hat{\beta} = \mathbf{r}$

where $\mathbf{C} = \mathbf{X}'\mathbf{X}$ (aka *the matrix of coefficients*) and $\mathbf{r} = \mathbf{X}'\mathbf{y}$ (aka the *right-hand-side*).

The above system involves p linear equations (each corresponding to one row in the system) of the form

$\Sigma_{k=1}^{p} C_{jk}\beta_k = r_j = \mathbf{C}[j,]'\hat{\beta} = \mathbf{r}[j]$

To illustrate the methods described below, I will use the *wages* data set

```
DATA=read.table('https://raw.githubusercontent.com/gdlc/STAT_COMP/master/DATA/wages.txt',header=TRUE,s

# using lm()
fm=lm(wage~education+sex+ethnicity,data=DATA)

X=as.matrix(model.matrix(~education+sex+ethnicity,data=DATA))
y=DATA$wage
XtX=crossprod(X)
Xty=crossprod(X,y)

bHat_1=solve(XtX)%*%Xty

# alternatively
bHat2=solve(XtX,Xty)

RES=data.frame("lm"=fm$coefficients)
RES
```

```
##                            lm
## (Intercept)       -3.11615191
## education          0.78776535
## sexMale            2.08858779
## ethnicityHispanic -0.09261815
## ethnicityOther     0.89019757
```

2) Solving the system using direct inversion

The first method we studied uses direct-inversion of the matrix of coefficients. We considered two variants of it

```r
bHat1=solve(XtX)%*%Xty

# alternatively
bHat2=solve(XtX,Xty)

RES$solve_1=bHat1
RES$solve_2=bHat2
RES
```

```
##                            lm     solve_1     solve_2
## (Intercept)       -3.11615191 -3.11615191 -3.11615191
## education          0.78776535  0.78776535  0.78776535
## sexMale            2.08858779  2.08858779  2.08858779
## ethnicityHispanic -0.09261815 -0.09261815 -0.09261815
## ethnicityOther     0.89019757  0.89019757  0.89019757
```

3) Using the QR-dcomposition

Recall:

- The QR decomposition $\mathbf{X} = \mathbf{Q_{n\times p}R_{p\times p}}$, where $\mathbf{Q'Q} = \mathbf{I}_p$ is an orthonormal matrix (i.e., an orthnormal basis to the column-space of X)
- $(AB)' = B'A'$

Using this we can see that

$\mathbf{X'X} = (\mathbf{QR})'(\mathbf{QR}) = \mathbf{R'Q'QR} = \mathbf{R'R}$

and

$\mathbf{X'y} = (\mathbf{QR})'\mathbf{y} = \mathbf{RQ'y}$

Therefore,

$\mathbf{X'X}\beta = \mathbf{X'y}$ [1]

can be expressed as

$\mathbf{R'R}\hat{\beta} = \mathbf{R'Q'y}$

Pre-multiplying both sides by $R'^{-1} = R^{-1'}$ we get

$\mathbf{R}\hat{\beta} = \mathbf{Q'y}$

Recall that $\mathbf{R}$ is a triangular matrix; therfore, the system can be easily solved using a back-solve recursive algorithm without using matrix inversion.

```r
QR=qr(X)
Q=qr.Q(QR)
R=qr.R(QR)

Qy=crossprod(Q,y)
bHatQR=backsolve(R,Qy)
RES$bHatQR=bHatQR
RES
```

```
##                           lm     solve_1     solve_2      bHatQR
## (Intercept)       -3.11615191 -3.11615191 -3.11615191 -3.11615191
## education          0.78776535  0.78776535  0.78776535  0.78776535
## sexMale            2.08858779  2.08858779  2.08858779  2.08858779
## ethnicityHispanic -0.09261815 -0.09261815 -0.09261815 -0.09261815
## ethnicityOther     0.89019757  0.89019757  0.89019757  0.89019757
```

4) Using the singular-value decomposition (SVD) of X

Recall that the Singluar Value Decomposition (SVD) of a matrix is $\mathbf{X} = \mathbf{UDV'}$ where $\mathbf{U'U} = I_n$ is an orthonormal basis for the row-space of $\mathbf{X}$, $\mathbf{V'V} = \mathbf{VV'} = I_p$ is an orthnormal basis for the column-space spanned by the columns of $\mathbf{X}$, and $\mathbf{D} = Diag\{d_j \geq 0\}$ is a diagonal matix with the singular-values of $\mathbf{X}$.

It follows that $\mathbf{X'X} = \mathbf{VDU'UDV'} = \mathbf{V\Psi V'}$ where $\mathbf{\Psi} = \mathbf{D'D} = Diag\{d_j^2\}$ is a diagonal matrix with the eigenvalues of $\mathbf{X'X}$ (the squared singular-values of $\mathbf{X}$) in the diagonal. This is known as the eigenvalue-decomposition of $\mathbf{X'X}$. Replacing in the sytem of OLS equations $\mathbf{X'X}$ with its with $\mathbf{V\Psi V'}$ $\mathbf{X}$ with it's singular-value decomposition ($\mathbf{X} = \mathbf{UDV'}$) we get,

$\mathbf{V\Psi V'}\hat{\beta} = \mathbf{VDU'y}$

Pre-multiplying both sides of the system by $\mathbf{V'}$ we get

$\mathbf{\Psi V'}\hat{\beta} = \mathbf{DU'y}$

Next, pre-multiplying both sides by $\mathbf{\Psi^{-1}}$ we get

$\mathbf{V'}\hat{\beta} = \mathbf{D^{-1}U'y}$

Finally, pre-multiplying both sides by $\mathbf{V}$ we get

$\hat{\beta} = \mathbf{VD^{-1}U'y}$

```r
SVD=svd(X)
U=SVD$u
DInv=diag(1/SVD$d)
V=SVD$v
bHatSVD=V%*%DInv%*%crossprod(U,y)
RES$bHatSVD=bHatSVD

RES
```

```
##                           lm     solve_1     solve_2      bHatQR     bHatSVD
## (Intercept)       -3.11615191 -3.11615191 -3.11615191 -3.11615191 -3.11615191
## education          0.78776535  0.78776535  0.78776535  0.78776535  0.78776535
## sexMale            2.08858779  2.08858779  2.08858779  2.08858779  2.08858779
## ethnicityHispanic -0.09261815 -0.09261815 -0.09261815 -0.09261815 -0.09261815
## ethnicityOther     0.89019757  0.89019757  0.89019757  0.89019757  0.89019757
```

5) Using the eigen-value decomposition (EVD) of X'X

```
EVD=eigen(XtX)
V=EVD$vectors
DInv=diag(1/EVD$values)
bHatEVD=V%*%DInv%*%t(V)%*%crossprod(X,y)
RES$bHatEVD=bHatEVD
RES
```

```
##                             lm     solve_1     solve_2      bHatQR     bHatSVD
## (Intercept)        -3.11615191 -3.11615191 -3.11615191 -3.11615191 -3.11615191
## education           0.78776535  0.78776535  0.78776535  0.78776535  0.78776535
## sexMale             2.08858779  2.08858779  2.08858779  2.08858779  2.08858779
## ethnicityHispanic -0.09261815 -0.09261815 -0.09261815 -0.09261815 -0.09261815
## ethnicityOther      0.89019757  0.89019757  0.89019757  0.89019757  0.89019757
##                        bHatEVD
## (Intercept)        -3.11615191
## education           0.78776535
## sexMale             2.08858779
## ethnicityHispanic -0.09261815
## ethnicityOther      0.89019757
```

6) Using the Gauss-Seidel algorithm

In the previous examples we showed how to sovle systems of linear equations using direct inversion and matrix factorizations (SVD, QR). Another alternative is to use an iterative algorithm.

Recall that the system

$\mathbf{C}\beta = \mathbf{r}$

involves p linear equations (each corresponding to one row in the system) of the form

$\Sigma_{k=1}^{p} C_{jk}\beta_k = r_j = \mathbf{C}[i,]'\hat{\beta} = \mathbf{r}[j]$

If we know all but the j^{th} coefficient, we can derive a solution for the uknown β_j coefficient using the j^{th} equation

$\hat{\beta}_j = \frac{r_j - \Sigma_{k \neq j} C_{jk}\beta_k}{C_{jj}}$ [2]

This suggest the following algorithm

Gauss-Seidel

- 1) Initialize all coefficients with some values (e.g., $\beta_j = 0 \ \forall j$)
- 2) In a loop (for $i \in 1, 2, ..., p$) solve for the jth coefficient using equation [2]. Replace the jth coefficient with the solution obtained.
- 3) Repeat (2) a large number of times, until the solution stablizes.