Cholesky Decomposition

Adam Smith, Spring 2017

1 Introduction

The problem of matrix inversion is a recurring theme in the computation associated with Bayesian models. For example, consider a Bayesian analysis of the regression model

$$y = X\beta + \varepsilon,\ \ \varepsilon \sim N(0, \sigma^2 I_n) \tag{1}$$

with priors defined as

$$\beta|\sigma^2 \sim N(\bar{\beta}, \sigma^2 A^{-1}) \tag{2}$$
$$\sigma^2 \sim \frac{\nu_0 s_0^2}{\chi^2_{\nu_0}}.$$

The conditional posterior of β given σ^2 is given by

$$\beta|\sigma^2, y \sim N(\tilde{\beta}, \sigma^2 (X'X + A)^{-1}) \tag{3}$$

where $\tilde{\beta} = (X'X + A)^{-1}(X'X\hat{\beta} + A\bar{\beta})$. Efficient sampling from $p(\beta|\sigma^2, y)$ critically depends on quick inversion methods for $(X'X + A)$.

2 Cholesky Decomposition and Matrix Inversion

2.1 Defining the Cholesky Decomposition

Let Σ be a symmetric positive-definite matrix. The Cholesky decomposition of Σ is defined as

$$\Sigma = U'U \tag{4}$$

where U denotes the upper triangular "Cholesky root" matrix. The following R code works through a simple example.

```
> Sigma
     [,1] [,2]
[1,]    2    1
[2,]    1    3
> U = chol(Sigma)
> U
```

```
          [,1]      [,2]
[1,] 1.414214 0.7071068
[2,] 0.000000 1.5811388
> t(U)%*%U
     [,1] [,2]
[1,]    2    1
[2,]    1    3
```

2.2 Matrix Inversion

Now consider the problem of inverting Σ. The simplest method is to use the `solve` function in R.

```
> solve(Sigma)
     [,1] [,2]
[1,]  0.6 -0.2
[2,] -0.2  0.4
```

However, a more efficient approach is available using the Cholesky decomposition of Σ.

Result 1 *If Σ is a symmetric positive-definite matrix with Cholesky decomposition $\Sigma = U'U$, then*

$$\Sigma^{-1} = (U^{-1})(U^{-1})'. \tag{5}$$

This result shows that the inverse of Σ can be computed only using the inverse of the Cholesky root U. That is, we have replaced the problem of inverting Σ with the problem of inverting U. The fact that U is upper triangular leads to faster and more numerically stable inversion methods relative to Σ.

The following R code uses the previous result to compute Σ^{-1}.

```
> U = chol(Sigma)
> U
          [,1]      [,2]
[1,] 1.414214 0.7071068
[2,] 0.000000 1.5811388

> IR = backsolve(U,diag(nrow(U)))
> IR
```

```
            [,1]       [,2]
[1,] 0.7071068 -0.3162278
[2,] 0.0000000  0.6324555
> IR%*%t(IR)
     [,1] [,2]
[1,]  0.6 -0.2
[2,] -0.2  0.4
```

Here `IR` refers to the "inverse (Cholesky) root" of Σ. Additionally, notice that `backsolve` is used in place of `solve` for computing `IR`. While `solve(U)` and `backsolve(U,diag(nrow(U)))` produce equivalent results, `backsolve` is preferred because it explicitly solves *triangular* systems of equations.

2.3 Application to the Bayesian Linear Model

Consider the problem of sampling from the posterior distribution of β defined in (3). Using the results of the previous section, we first write

$$\Sigma = (X'X + A) = U'U \tag{6}$$

where U is the upper triangular Cholesky root of $(X'X + A)$. It follows that

$$\Sigma^{-1} = (X'X + A)^{-1} = (U^{-1})(U^{-1})' = (IR)(IR)'. \tag{7}$$

The following R code generates one draw of β from its posterior conditional on σ^2.

```
U = chol(crossprod(X)+A)
IR = backsolve(U,diag(nrow(U)))
btilde = crossprod(t(IR))%*%(crossprod(X)%*%bhat + A%*%betabar)
beta = btilde + sqrt(sigmasq)*IR%*%rnorm(nvar)
```

References

Rossi, P. E., G. M. Allenby, and R. McCulloch (2005), *Bayesian Statistics and Marketing*. New York: John Wiley and Sons.