Descomposición (factorización) de Cholesky 2

Análisis Numérico

Joaquin Cavieres G.

Eficiencia en el calculo de matrices

Las matrices son utilizadas en la gran parte de los análisis estadísticos (por no decir siempre). Sin embargo, los modelos estadísticos con una gran cantidad de observaciones, sirven cuando estos pueden ser "traspasados" a la computadoras y hacer los calculos de forma eficiente. Generalmente es facíl arruinar la teoría matemática con una pobre implementación del álgebra lineal y su calculo numérico.

Calculo con matrices

Considere el siguiente ejemplo:

```
n =  1500                    # N° de observaciones
A =  matrix(runif(n*n),n,n)  # Matriz A de dimensión n x n
B =  matrix(runif(n*n),n,n)  # matriz B de dimensión n x n
y =  runif(n)                # Vector y de dimensión n
```

Caso 1

```
library(tictoc)
tic("Tiempo de calculo")
caso1 = A%*%B%*%y
toc()
```

```
## Tiempo de calculo: 2.31 sec elapsed
```

Caso 2

```
library(tictoc)
tic("Tiempo de calculo")
caso2 = A%*%(B%*%y)
toc()
```

```
## Tiempo de calculo: 0 sec elapsed
```

¿Por que se dan estas diferencias? La respuesta es en como se hicieron las multiplicaciones y el número de operaciones de punto flotante (*flop*) requeridos para realizar estos calculos.

Veamos con más detalle cada operación:

- En la primera operación, la matriz $\boldsymbol{A}$ y $\boldsymbol{B}$ se crearon en primera instancia para luego ser multiplicadas por el vector $\boldsymbol{y}$.
- En la segunda operación primero se realizó la operación $\boldsymbol{By}$ y luego multiplicado por la matriz $\boldsymbol{A}$.

¿Que son los *flop*?

Las operaciones de punto flotante por segundo son una medida del rendimiento de una computadora, especialmente en cálculos científicos que requieren un gran uso de operaciones de punto flotante.

Ejemplo: Considere el calculo de la traza de la multiplicación de la matriz `tr(AB)` con $\boldsymbol{A}$ de dimensión 2000×100 y y $\boldsymbol{B}$ de dimensión 100×2000.

```
n = 5000
m = 100
A = matrix(runif(n*m),n,m)
B = matrix(runif(n*m),m,n)

library(tictoc)
tic("Tiempo de calculo")
caso1 = sum(diag(A%*%B))
toc()
```

```
## Tiempo de calculo: 1.25 sec elapsed
```

```
library(tictoc)
tic("Tiempo de calculo")
caso2 = sum(diag(B%*%A))
toc()
```

```
## Tiempo de calculo: 0.03 sec elapsed
```

```
library(tictoc)
tic("Tiempo de calculo")
caso3 = sum(A*t(B))
toc()
```

```
## Tiempo de calculo: 0 sec elapsed
```

En el primer caso (`caso1`), el calculo de $\boldsymbol{AB}$ tiene un costo de *flops* es de $2n^2m$ y luego se extrae la diagonal y la suma. En el segundo caso (`caso2`) utiliza `tr(AB)` $=$ `tr(BA)`. Lo anterior es matemáticamente equivalente y el costo de multiplicar $\boldsymbol{BA}$ es de $2nm^2$, para luego estraer la diagnoal y hacer la suma. En el tercer caso (`caso3`) se utiliza la forma directa de la traza `tr(AB)` $= \sum_{ij} A_{ij}B_{ji}$

con costo de operación $2nm$. Este caso es el más rápido en la estimación ya que no se desperdicia ningún esfuerzo en calcular los elementos de la diagonal de la matriz objetivo.

Los *flops* son importante cuando queremos hacer calculos en el computador, pero también se debe tener en cuenta que no es sencillo hacer un recuento completo de los *flops* para algoritmos complejos. En muchos casos no es necesario un recuento exacto y simplemente se considera escalar el cálculo de los *flops* con el tamaño del problema. Por esta razón, el recuento de operaciones principales es todo lo que se reporta generalmente.

Descomposición de Cholesky

Sin pensamos en un símil de las matrices definidas positivas a lo que nosotros ya conocemos, entonces podemos decir que son los "números reales positivos" del álgebra matricial. Tienen ventajas computacionales particulares y generalmente son utilizadas en la estadística, ya que las matrices de covarianza suelen ser definidas positivas (y siempre semidefinidas positivas). Por tanto, veamos un ejemplo de matrices definidas positivas y sus raíces cuadradas de matriz. Para ver por qué las raíces cuadradas de la matriz pueden ser útiles, considere lo siguiente.

Ejemplo: Si simulamos una variable aleatoria $\boldsymbol{y}$ multivariante *i.i.d*, esto sería $\boldsymbol{y} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Nosotros podríamos generalizar variables aleatorias $\boldsymbol{z}$ desde $\boldsymbol{y} \sim N(\boldsymbol{0}, \boldsymbol{I})$ y, a la vez, podriamos encontrar una matriz $\boldsymbol{R}$ tal que $\boldsymbol{R}^T\boldsymbol{R} = \boldsymbol{\Sigma}$. Así:

$$\boldsymbol{y} \equiv \boldsymbol{R}^T\boldsymbol{z} + \boldsymbol{\mu} \sim N(\mu, \boldsymbol{\Sigma})$$

ya que la matriz de covarianza de $\boldsymbol{y} = \boldsymbol{R}^T\boldsymbol{I}\boldsymbol{R} = \boldsymbol{R}^T\boldsymbol{R} = \boldsymbol{\Sigma}$ y $\mathbb{E}(\boldsymbol{y}) = \mathbb{E}(\boldsymbol{R}^T\boldsymbol{z} + \boldsymbol{\mu}) = \boldsymbol{\mu}$.

$$\boldsymbol{y} = \left(\begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}, \begin{pmatrix} 2 & -1 & 1 \\ -1 & 2 & -1 \\ 1 & -1 & 2 \end{pmatrix} \right)$$

```
Cov_mat = matrix(c(2,-1,1,-1,2,-1,1,-1,2),3,3)
mu = c(1,-1,3)
R = chol(Cov_mat)   # Choleksy sobre 'Cov_mat'
z = rnorm(3)       ## 3 iid N(0,1) observaciones

y = t(R)%*%z + mu  ## N(mu,Cov_mat) deviates
y
```

```
##           [,1]
## [1,]  3.557471
## [2,] -2.504309
## [3,]  5.544123
```

A continuación, se simulan 500 de estos vectores y se comprueba su media y covarianza observadas.

```r
Z = matrix(rnorm(1500),3, 500) # 500 N(0,I) 3 vectores
Y = t(R)%*%Z + mu              # 500 N(mu,V) vectores
```

y vemos el comportamiento de ellos:

```r
rowMeans(Y)                    # Mu observado
```

```
## [1]  0.9525481 -1.0107243  2.9808623
```

```r
(Y-mu)%*%t(Y-mu)/500           # Cov_mat observada
```

```
##            [,1]      [,2]      [,3]
## [1,]  2.154901 -1.252417  1.212596
## [2,] -1.252417  2.296684 -1.249295
## [3,]  1.212596 -1.249295  2.154731
```