Álgebra Linear

Prof. Wagner Hugo Bonat

1. A decomposição de Cholesky de uma matriz simétrica e definida positiva é muito popular em estatística. Para você ter uma ideia do algoritmo consulte. Use três diferentes abordagens para obter a decomposição de Cholesky de uma matriz positiva definida em `R` e/ou `C++`. As abordagens podem ser diferentes pacotes, diferentes classes ou mesmo diferentes linguagens. Se você conhece outras linguagens pode usar se julgar adequado. Para criar uma matriz positiva definida use o seguinte código. Tome cuidado com a classe das matrizes que você vai utilizar para comparar as diferentes abordagens. Considere matrizes de diferentes dimensões e use o pacote `bench` para a comparação em termos de tempo computacional. Importante explique cuidadosamente a diferença entre as abordagens e qual você julga ser a mais eficiente antes e após realizar o experimento computacional.

```
x1 <- runif(30)
x2 <- runif(30)
grid <- expand.grid(x1, x2)
DD <- dist(grid, diag = TRUE, upper = TRUE)
DD_positiva <- exp(as.matrix(-DD, 100, 100)/0.3)
```

2. Nas mesmas condições do exercício 1. Considere que é de interesse obter a decomposição em autovalores e autovetores. Novamente forneça três alternativas e compare os tempos computacionais.

3. Considere um modelo linear de covariancia com matriz de covariância descrita por dois componentes conforme código abaixo. A (i, j)-ésima entrada da matriz de sensitividade para estimação dos parâmetros de dispersão é dada por
$$S_{\tau_{ij}} = -\text{tr}(W_{\tau_i}\mathbf{C}W_{\tau_j}C),$$
onde $C = \tau_1\text{I} + \tau_2 Z$ e $W_{\tau_i} = \frac{\partial C}{\partial \tau_i}$ para $i = 1, 2$. Note que no caso de dois parâmetros $(\tau_1, \tau_2)^\top$ a matriz de sensitividade é 2×2. Proponha três estratégias para obter a matriz de sensitivade. Compare suas propostas pelo tempo computacional. Avalie matrizes de diferentes tamanhos. Considere que a matriz Z é bloco diagonal com estrutura dada pelo código abaixo.

```
library(Matrix)
Z_temp <- rep(1, 5)%*%t(rep(1, 5))
Z_lista <- list()
for(i in 1:10) {Z_lista[[i]] <- Z_temp}
Z <- Matrix::bdiag(Z_lista)
C <- 5*Diagonal(50, 1) + 3*Z
```

4. A distribuição Normal multivariada tem uma ampla gama de aplicações em estatística. Exemplos incluem análise de componentes principais, regressão multivariada, análise de variância multivariada, análise fatorial entre outras. Dizemos que um vetor aleatório Y de dimensão $n \times 1$ tem distribuição normal multivariada se sua função densidade probabilidade é dada por

$$f(\mathbf{y}|\mu, \Sigma) = \left(\frac{1}{2\pi}\right)^{n/2} |\Sigma|^{-1/2} \exp\{-\frac{1}{2}(\mathbf{y} - \mu)^\top \Sigma^{-1}(\mathbf{y} - \mu)\}, \quad (1)$$

onde μ é um vetor $n \times 1$ de valores esperados e Σ é uma matriz $(n \times n)$ simétrica e positiva definida. É usual representar esta situação pela notação $Y \sim N(\mu, \Sigma)$. E neste caso temos que $E(Y) = \mu$ e $V(Y) = \Sigma$.

Implemente esta função usando pelo menos três diferentes abordagens de álgebra linear. Note que a implementação desta função depende de três componentes chaves: i) Cálculo do determinante de Σ; ii) Multiplicação de matrizes e iii) Inversa da matriz Σ. Você pode usar qualquer estratégia que achar conveniente relacionado a propriedades de matrizes e/ou resolução de sistemas lineares. Você precisará especificar o vetor μ para o qual sugiro usar um vetor de zeros e para a matriz Σ que deve ser positiva definida. Considere vetores de tamanho entre 10 e 100.

O pacote **mvtnorm** do software `R` forcene uma implementação de tal distribuição através da função `dmvnorm()`. Use esta implementação como base de comparação e faça com que sua função seja mais rápida. O código abaixo ilustra o uso do pacote para avaliar a distribuição normal multivariada.

```
require(mvtnorm)
```

```
## Carregando pacotes exigidos: mvtnorm
```

```
# Loading required package: mvtnorm
mu <- c(0,0)
Sigma <- matrix(c(1,0.8,0.8,1),2,2)
dmvnorm(x = c(0,0), mean = mu, sigma = Sigma)
```

```
## [1] 0.2652582
```

5. O seguinte post clique aqui apresenta várias considerações relacionadas a performance computacional de cálculos matriciais envolvendo matrizes esparsas com o Rcpp e Armadillo. Leia o artigo, reproduza o código e faça um resumo do que o post apresenta.