Khyati Naik: Data 605 - HW2

Sep 10, 2023

Contents

Problem Set 1

(1) Show that ATA != AAT in general. (Proof and demonstration.)

```
# Create a random square matrix A
set.seed(123)  # For reproducibility
n <- 3  # Change this to the size of the matrix
A <- matrix(rnorm(n^2), nrow = n)

# Display the original matrix A
cat("Matrix A:\n")
```

```
## Matrix A:
```

```
print(A)
```

```
##            [,1]       [,2]       [,3]
## [1,] -0.5604756 0.07050839  0.4609162
## [2,] -0.2301775 0.12928774 -1.2650612
## [3,]  1.5587083 1.71506499 -0.6868529
```

```
# Calculate AT*A and AAT
ATA <- t(A) %*% A

# Display the ATA
cat("Matrix ATA:\n")
```

```
## Matrix ATA:
```

```
print(ATA)
```

```
##           [,1]      [,2]      [,3]
## [1,]  2.796686  2.604009 -1.037747
## [2,]  2.604009  2.963135 -1.309056
## [3,] -1.037747 -1.309056  2.284591
```

```r
AAT <- A %*% t(A)

# Display the AAT
cat("Matrix AAT:\n")
```

```
## Matrix AAT:
```

```r
print(AAT)
```

```
##            [,1]       [,2]       [,3]
## [1,]  0.5315481 -0.4449625 -1.0692732
## [2,] -0.4449625  1.6700769  0.7318682
## [3,] -1.0692732  0.7318682  5.8427864
```

```r
# Check if AT*A is not equal to AAT in general
not_equal <- any(ATA != AAT)

# Display results
if (not_equal) {
  cat("AT * A is not equal to A * AT\n")
} else {
  cat("AT * A is equal to A * AT\n")
}
```

```
## AT * A is not equal to A * AT
```

(2) For a special type of square matrix A, we get AT = AAT. Under what conditions could this be true? (Hint: The Identity matrix I is an example of such a matrix)

```r
# Generate a random orthogonal matrix using SVD
set.seed(123)
n <- 3  # Change this to the size of the matrix
A <- matrix(rnorm(n^2), nrow = n)
U <- svd(A)$u

# Calculate A^T * A and A * A^T
ATA <- t(U) %*% U
AAT <- U %*% t(U)

# Check if A^T * A is equal to A * A^T
equal_condition <- all.equal(ATA, AAT)

# Display results
cat("Matrix U (Orthogonal Matrix):\n")
```

```
## Matrix U (Orthogonal Matrix):
```

```r
print(U)
```

```
##             [,1]       [,2]      [,3]
## [1,] -0.1960019  0.1876318 0.9624851
## [2,]  0.1753957 -0.9589867 0.2226676
## [3,]  0.9647899  0.2124590 0.1550535
```

```
if (equal_condition) {
  cat("\nA^T * A is equal to A * A^T for orthogonal matrix A\n")
} else {
  cat("\nA^T * A is not equal to A * A^T for orthogonal matrix A\n")
}
```

```
##
## A^T * A is equal to A * A^T for orthogonal matrix A
```

```
# Verify A^T * A = I
identity_matrix <- diag(n)
equal_identity <- all.equal(ATA, identity_matrix)

if (equal_identity) {
  cat("\nA^T * A is equal to the identity matrix I\n")
} else {
  cat("\nA^T * A is not equal to the identity matrix I\n")
}
```

```
##
## A^T * A is equal to the identity matrix I
```

Problem Set 2

Write an R function to factorize a square matrix A into LU or LDU, whichever you prefer.

```
LU_decomposition <- function(A) {
  n <- dim(A)[1]
  U <- A
  L <- diag(n)

  if (n == 1) {
    return(list(L = L, U = U))
  }

  for (i in 2:n) {
    for (j in 1:(i - 1)) {
      multiplier <- -U[i, j] / U[j, j]
      U[i, ] <- multiplier * U[j, ] + U[i, ]
      L[i, j] <- -multiplier
    }
  }

  return(list(L = L, U = U))
}

# Test the LU decomposition function
```

```r
A <- matrix(c(2, -1, -2, 4, 1, -3, 0, 3, -2), nrow = 3)
LU <- LU_decomposition(A)

cat("Matrix A:\n")
```

```
## Matrix A:
```

```r
print(A)
```

```
##      [,1] [,2] [,3]
## [1,]    2    4    0
## [2,]   -1    1    3
## [3,]   -2   -3   -2
```

```r
cat("Matrix L:\n")
```

```
## Matrix L:
```

```r
print(LU$L)
```

```
##      [,1]      [,2] [,3]
## [1,]  1.0 0.0000000    0
## [2,] -0.5 1.0000000    0
## [3,] -1.0 0.3333333    1
```

```r
cat("Matrix U:\n")
```

```
## Matrix U:
```

```r
print(LU$U)
```

```
##      [,1] [,2] [,3]
## [1,]    2    4    0
## [2,]    0    3    3
## [3,]    0    0   -3
```

```r
A == LU$L %*% LU$U
```

```
##      [,1] [,2] [,3]
## [1,] TRUE TRUE TRUE
## [2,] TRUE TRUE TRUE
## [3,] TRUE TRUE TRUE
```