

Algoritmo Strassen para multiplicación de matrices

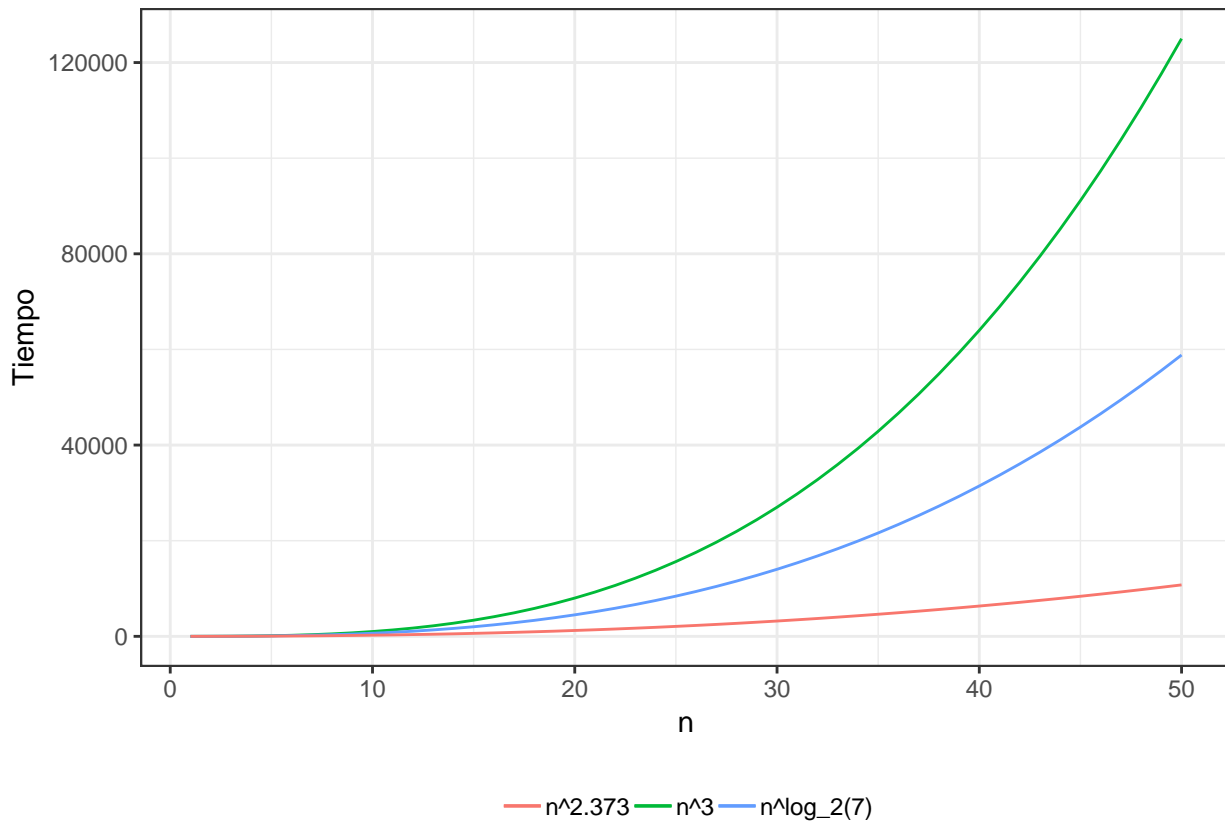
Gabriela Vargas, Dalia Camacho, Elizabeth Monroy

El método usual para multiplicar matrices es de orden $O(n^3)$, sin embargo en 1969 Volker Strassen propuso un método más eficiente para la multiplicación de matrices cuadradas de tamaño $n \times n$, donde n es par. Este método consiste en dividir las dos matrices en cuatro submatrices de tamaño $\frac{n}{2}$; realizar operaciones aditivas de estas matrices; llevar a cabo 7 multiplicaciones de matrices y finalmente unir los resultados. De esta forma la multiplicación de matrices se convierte en un problema de orden $O(n^{\log 7}) = O(n^{2.81})$ superando asíntoticamente al método clásico de orden $O(n^{\log 8}) = O(n^3)$.

Como ejemplo particular para matrices de 2×2 con el método de Strassen se requieren 7 multiplicaciones escalares, en vez de las 8 necesarias en el método clásico.

El algoritmo de Strassen hace uso del método *Divide y vencerás*, esto implica resolver un problema difícil dividiéndolo en partes tantas veces como sea necesario hasta que la solución de las partes sea simple.

Otros métodos se han desarrollado para disminuir el orden del algoritmo de multiplicación de matrices. El algoritmo más eficiente hasta ahora es el de Virginia Vassilevska Williams, quien en 2014 logró bajar el orden a $O(n^{2.373})$. Este algoritmo utiliza formas trilineales, en vez de bilineales para representar la multiplicación de matrices.



Dadas dos matrices cuadradas A y B , de tamaño $n \times n$, el producto de matrices C se define como

$$\begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \cdot \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}$$

Aplicando el método tradicional de multiplicación de matrices, tenemos que:

$$\begin{aligned}
C_{11} &= A_{11} \cdot B_{11} + A_{12} \cdot B_{12}, \\
C_{12} &= A_{11} \cdot B_{12} + A_{12} \cdot B_{22}, \\
C_{21} &= A_{21} \cdot B_{11} + A_{22} \cdot B_{21}, \\
C_{22} &= A_{21} \cdot B_{12} + A_{22} \cdot B_{22}
\end{aligned} \tag{1}$$

En cada una de las cuatro ecuaciones se especifican dos multiplicaciones de matrices de tamaño $n/2 \times n/2$ y la suma de estos productos.

La idea del algoritmo Strassen es particionar las matrices A y B tal que en lugar de ocupar 8 multiplicaciones para obtener la matriz $A \cdot B = C$, únicamente se ocupen 7. Este algoritmo consta de cuatro pasos:

1. Dividir las matrices A , B y C en submatrices de tamaño $n/2 \times n/2$.
2. Crear 10 matrices S_1, S_2, \dots, S_{10} , de $n/2 \times n/2$ cada una, las cuales son resultado de la suma o resta de las matrices creadas en el paso uno.
3. Calcular 7 productos de matrices P_1, P_2, \dots, P_7 mediante las submatrices creadas en el paso uno y las submatrices S creadas en el paso dos.
4. Calcular las submatrices $C_{11}, C_{12}, C_{21}, C_{22}$ a través de la suma y resta de diferentes combinaciones de las matrices P_i .

Las diez matrices del paso dos están definidas como:

$$\begin{aligned}
S_1 &= B_{12} - B_{22} \\
S_2 &= A_{11} + A_{12} \\
S_3 &= A_{21} + A_{22} \\
S_4 &= B_{21} - B_{11} \\
S_5 &= A_{11} + A_{22} \\
S_6 &= B_{11} + B_{22} \\
S_7 &= A_{12} - A_{22} \\
S_8 &= B_{21} + B_{22} \\
S_9 &= A_{11} - A_{21} \\
S_{10} &= B_{11} + B_{12}.
\end{aligned} \tag{2}$$

Las multiplicaciones de matrices del paso tres están dadas por:

$$\begin{aligned}
P_1 &= A_{11} \cdot S_1 \\
P_2 &= S_2 \cdot B_{22} \\
P_3 &= S_3 \cdot B_{11} \\
P_4 &= A_{22} \cdot S_4 \\
P_5 &= S_5 \cdot S_6 \\
P_6 &= S_7 \cdot S_8 \\
P_7 &= S_9 \cdot S_{10}.
\end{aligned} \tag{3}$$

Para obtener la matriz $C = A \cdot B$ calculamos las submatrices $C_{11}, C_{12}, C_{21}, C_{22}$:

$$\begin{aligned}
C_{11} &= P_5 + P_4 - P_2 + P_6 \\
C_{12} &= P_1 + P_2 \\
C_{21} &= P_3 + P_4 \\
C_{22} &= P_5 + P_1 - P_3 - P_7.
\end{aligned} \tag{4}$$

Implementación

Se define una función para ejecutar el algoritmo.

```
# La función divide recibe como input una matriz cuadrada  
# A de tamaño nxn.  
# El output de la función es una lista con las cuatro  
# submatrices de tamaño n/2xn/2.
```

```
divide <- function(A){  
  r <- dim(A)[1]  
  A <- list(A)  
  A11 <- A[[1]][1:(r/2),1:(r/2)]  
  A12 <- A[[1]][1:(r/2),(r/2+1):r]  
  A21 <- A[[1]][(r/2+1):r,1:(r/2)]  
  A22 <- A[[1]][(r/2+1):r,(r/2+1):r]  
  A <- list(X11=A11, X12=A12, X21=A21, X22=A22)  
  return(A)  
}
```

```
# El código de Strassen con cada paso en el algoritmo  
# por separado
```

```
strassen_largo <- function(A,B){  
  # Dividir las matrices A y B en matrices de n/2xn/2  
  A <- divide(A)  
  B <- divide(B)  
  
  # Generar las diez matrices del algoritmo de Strassen  
  S1 <- B$X12 - B$X22  
  S2 <- A$X11 + A$X12  
  S3 <- A$X21 + A$X22  
  S4 <- B$X21 - B$X11  
  S5 <- A$X11 + A$X22  
  S6 <- B$X11 + B$X22  
  S7 <- A$X12 - A$X22  
  S8 <- B$X21 + B$X22  
  S9 <- A$X11 - A$X21  
  S10 <- B$X11 + B$X12
```

```
# Se calculan los productos que dan lugar a las  
# 7 matrices P
```

```
P1 <- A$X11 %*% S1  
P2 <- S2 %*% B$X22  
P3 <- S3 %*% B$X11  
P4 <- A$X22 %*% S4  
P5 <- S5 %*% S6  
P6 <- S7 %*% S8  
P7 <- S9 %*% S10
```

```
# Se suman y restan las matrices P para obtener la  
# multiplicación de A y B por subcuadrante
```

```
C11 <- P5 + P4 - P2 + P6  
C12 <- P1 + P2
```

```

C21 <- P3 + P4
C22 <- P5 + P1 - P3 - P7

# Juntar los subcuadrantes en la matriz C
C <- rbind(cbind(C11,C12),cbind(C21,C22))

return(C)
}

# El código de Strassen uniendo los pasos 2 y 3 en uno solo

strassen <- function(A,B){
  # Dividir las matrices A y B
  A <- divide(A)
  B <- divide(B)

  # Juntar los pasos 2 y 3

  M1 <- (A$X11+A$X22)%*(B$X11+B$X22)
  M2 <- (A$X21+A$X22)%*B$X11
  M3 <- A$X11%*(B$X12-B$X22)
  M4 <- (A$X22)%*(B$X21-B$X11)
  M5 <- (A$X11+A$X12)%*B$X22
  M6 <- (A$X21-A$X11)%*(B$X11+B$X12)
  M7 <- (A$X12-A$X22)%*(B$X21+B$X22)

  # Formar la matriz C con las Ms calculadas previamente
  C11<-M1+M4-M5+M7
  C12<-M3+M5
  C21<-M2+M4
  C22<-M1-M2+M3+M6

  # Juntar los subcuadrantes en la matriz C
  C<-rbind(cbind(C11,C12),cbind(C21,C22))

  return(C)
}

```

Se prueba el algoritmo para calcular la multiplicación de las matrices A y B por medio del algoritmo Strassen. Se compara el resultado con la matriz C que representa el resultado de la operación de producto matricial en R.

```

set.seed(53564)
# Crear matrices para la prueba
A <- matrix(round(runif(64,0,1)*10),8)
A

##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
## [1,]    6    5    3    4    9    6    5    3
## [2,]    7    9    3    9    4    6    3    3
## [3,]    2    3    2    2    2    1    7    8

```

```
## [4,] 8 3 6 6 8 7 2 0
## [5,] 7 1 9 4 10 0 1 6
## [6,] 10 7 8 4 5 5 6 1
## [7,] 4 7 6 8 6 7 2 8
## [8,] 7 2 3 7 7 3 7 3
```

```
B <- matrix(round(runif(64,0,1)*10),8)
print("Matriz B")
```

```
## [1] "Matriz B"
```

```
print(B)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
## [1,] 1 7 2 4 1 2 3 7
## [2,] 3 4 6 8 3 8 0 1
## [3,] 3 5 5 3 0 7 4 2
## [4,] 1 3 4 5 9 6 3 4
## [5,] 7 9 9 5 1 2 2 1
## [6,] 9 0 4 9 9 5 4 2
## [7,] 9 9 3 8 7 4 2 2
## [8,] 0 6 2 6 6 7 7 4
```

Resultado de multiplicación de matrices usando el operador de R

```
C <- A%*%B
C
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
## [1,] 196 233 199 250 173 186 115 112
## [2,] 161 208 194 270 212 232 119 134
## [3,] 105 171 99 171 137 147 98 79
## [4,] 178 206 194 223 156 177 114 121
## [5,] 120 245 186 177 99 175 133 120
## [6,] 193 255 203 264 165 222 123 140
## [7,] 174 230 216 287 228 265 160 135
## [8,] 168 237 171 224 177 171 115 124
```

Resultado de multiplicación de matrices usando la función Strassen con los pasos dos y tres por separado:

```
C_strassenL <- strassen_largo(A,B)
```

```
C_strassenL
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
## [1,] 196 233 199 250 173 186 115 112
## [2,] 161 208 194 270 212 232 119 134
## [3,] 105 171 99 171 137 147 98 79
## [4,] 178 206 194 223 156 177 114 121
## [5,] 120 245 186 177 99 175 133 120
## [6,] 193 255 203 264 165 222 123 140
## [7,] 174 230 216 287 228 265 160 135
## [8,] 168 237 171 224 177 171 115 124
```

Comparación del método de R con Strassen con los pasos dos y tres por separado.

```
C_strassenL == C
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
## [1,] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

```
## [2,] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [3,] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [4,] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [5,] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [6,] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [7,] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [8,] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

Resultado de multiplicación de matrices, usando la función Strassen con los pasos dos y tres juntos:

```
C_strassenS <- strassen(A,B)
```

```
C_strassenS
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
## [1,] 196 233 199 250 173 186 115 112
## [2,] 161 208 194 270 212 232 119 134
## [3,] 105 171 99 171 137 147 98 79
## [4,] 178 206 194 223 156 177 114 121
## [5,] 120 245 186 177 99 175 133 120
## [6,] 193 255 203 264 165 222 123 140
## [7,] 174 230 216 287 228 265 160 135
## [8,] 168 237 171 224 177 171 115 124
```

Comparación del método de R con Strassen con los pasos dos y tres juntos.

```
C_strassenS == C
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
## [1,] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [2,] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [3,] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [4,] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [5,] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [6,] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [7,] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [8,] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```