Homework 0

Saulo Mendes

December 2020

Kronecker Product

Problem 1 For a randomly generated $\mathbf{A} \in \mathbb{C}^{N \times N}$ and $\mathbf{B} \in \mathbb{C}^{N \times N}$, evaluate the computational performance (run time) of the following matrix inversion formulas:

(a) for $N \in \{2, 4, 8, 16, 32, 64\}$

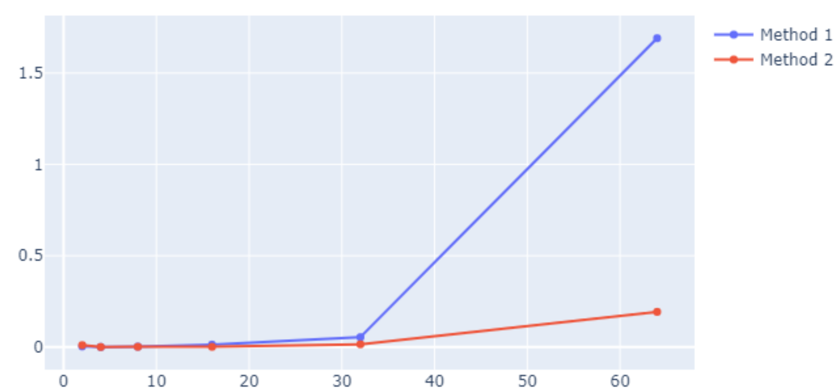
- Method 1: $(\mathbf{A}_{N \times N} \otimes \mathbf{B}_{N \times N})^{-1}$
  Solution:

```
N_list = [2, 4, 8, 16, 32, 64]
Method1_times = []

for n in N_list:
    A = randn(n, n)
    B = randn(n, n)

    start_time = time.time()

    A_x_B = kron(A, B)

    A_x_B_inv = inv(A_x_B)

    end_time = time.time()

    Method1_times.append(end_time-start_time)
```

- Method 2: $\mathbf{A}^{-1}_{N \times N} \otimes \mathbf{B}^{-1}_{N \times N}$
  Solution:

```python
N_list = [2, 4, 8, 16, 32, 64]
Method2_times = []

for n in N_list:
    A = randn(n, n)
    B = randn(n, n)

    start_time = time.time()

    A_inv = inv(A)
    B_inv = inv(B)

    A_x_B = kron(A, B)

    end_time = time.time()

    Method2_times.append(end_time-start_time)
```

- Comparison of results
  The graph below shows the run time difference between the methods. We can see a dramatically higher run time in higher dimensions of matrices in Method 1. This is due to the fact that with method one, we are calculating matrix inverses after the Kronecker product. For two $N \times N$ matrices as input, a Kronecker product produces a resulting matrix of $N^2 \times N^2$ dimension, which takes a much longer time to calculate the inverse.



(b) for $K \in \{2, 4, 6, 8, 10\}$

- Method 1: $\left(\mathbf{A}_{2\times 2}^{(1)} \otimes \mathbf{A}_{2\times 2}^{(2)} \otimes ... \otimes \mathbf{A}_{2\times 2}^{(K)}\right)^{-1} = \left(\otimes_{i=1}^{K} \mathbf{A}_{2\times 2}^{(i)}\right)^{-1}$

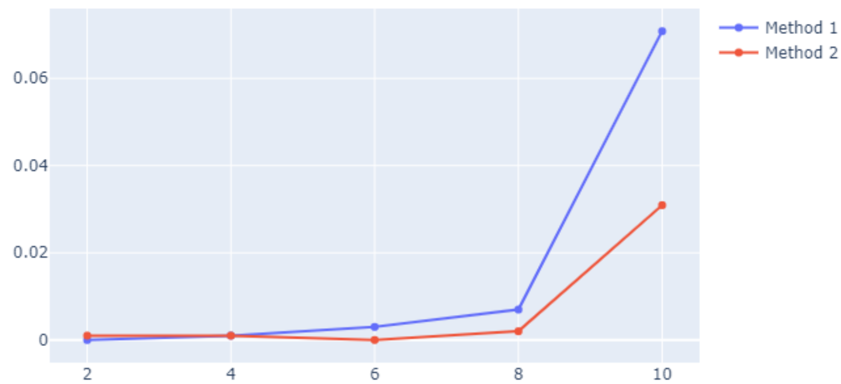
Solution:

```
K_list = [2, 4, 6, 8, 10]
Method1_times = []

for k_value in K_list:

    A_start = randn(2, 2)

    start_time = time.time()
    for k in range(0, k_value):
        if k == 0:
            A = A_start
        else:
            A = kron(A, A_start)

    A = inv(A)
    end_time = time.time()

    Method1_times.append(end_time-start_time)
```

- Method 2: $\left(\mathbf{A}^{(1)}\right)^{-1}_{2\times2} \otimes \left(\mathbf{A}^{(2)}\right)^{-1}_{2\times2} \otimes \ldots \otimes \left(\mathbf{A}^{(K)}\right)^{-1}_{2\times2} = \otimes_{i=1}^{K} \left(\mathbf{A}^{(i)}\right)^{-1}_{2\times2}$
  Solution:

```
K_list = [2, 4, 6, 8, 10]
Method2_times = []

for k_value in K_list:

    A_start = randn(2, 2)

    start_time = time.time()
    for k in range(0, k_value):
        if k == 0:
            A = inv(A_start)
        else:
            A = kron(A, inv(A_start))

    end_time = time.time()

    Method2_times.append(end_time-start_time)
```

- Comparison of results
  The graph below shows the run time difference between the methods. Similarly to the previous problem, here method 2 also has a shorter run time due the fact that the matrix inverse calculations are being done in lower dimension matrices, rather than the larger ones generated by the Kronecker product.



Problem 2 Let $eig(\mathbf{X})$ be the function that returns the matrix $\mathbf{\Sigma}_{K\times K}$ of eigenvalues of $\mathbf{X}$. Show algebraically that $eig(\mathbf{A}\otimes\mathbf{B}) = eig(\mathbf{A})\otimes eig(\mathbf{B})$.

<u>Hint</u>: Use the property

$$(\mathbf{A}\otimes\mathbf{B})(\mathbf{C}\otimes\mathbf{D}) = \mathbf{AC}\otimes\mathbf{BD} \tag{1}$$

Solution: Let $\mathbf{A}\in\mathbb{C}^{M\times M}$ with eigenvector matrix $\mathbf{V}$ and eigenvalue matrix $\mathbf{\Sigma}_A$, and $\mathbf{B}\in\mathbb{C}^{N\times N}$ with eigenvector matrix $\mathbf{U}$ and eigenvalue matrix $\mathbf{\Sigma}_B$. Their respective eigendecompositions are:

$$\mathbf{AV} = \mathbf{V\Sigma}_A \rightarrow \mathbf{A} = \mathbf{V\Sigma}_A\mathbf{V}^{-1} \tag{2}$$

$$\mathbf{BU} = \mathbf{U\Sigma}_B \rightarrow \mathbf{B} = \mathbf{U\Sigma}_B\mathbf{U}^{-1} \tag{3}$$

Taking the Kronecker product of $\mathbf{A}$ and $\mathbf{B}$ and applying property (1) and substitutions (2) and (3)

$$\begin{aligned}\mathbf{A}\otimes\mathbf{B} &= (\mathbf{V\Sigma}_A\mathbf{V}^{-1})\otimes(\mathbf{U\Sigma}_B\mathbf{U}^{-1})\\ &= (\mathbf{V\Sigma}_A\otimes\mathbf{U\Sigma}_B)(\mathbf{V}^{-1}\otimes\mathbf{U}^{-1})\\ &= (\mathbf{V}\otimes\mathbf{U})(\mathbf{\Sigma}_A\otimes\mathbf{\Sigma}_B)(\mathbf{V}^{-1}\otimes\mathbf{U}^{-1})\end{aligned}$$

We can make use of the property $(\mathbf{A} \otimes \mathbf{B})^{-1} = \mathbf{A}^{-1} \otimes \mathbf{B}^{-1}$ to have

$$\mathbf{A} \otimes \mathbf{B} = (\mathbf{V} \otimes \mathbf{U})(\mathbf{\Sigma}_A \otimes \mathbf{\Sigma}_B)(\mathbf{V} \otimes \mathbf{U})^{-1}$$

This is the eigendecomposition of $\mathbf{A} \otimes \mathbf{B}$, where $(\mathbf{\Sigma}_A \otimes \mathbf{\Sigma}_B)$ is its eigenvalue matrix $\mathbf{\Sigma}_{\mathbf{A} \otimes \mathbf{B}}$. Thus, $eig(\mathbf{A} \otimes \mathbf{B}) = eig(\mathbf{A}) \otimes eig(\mathbf{B})$.