Computer Homework 3

Matrix and Tensor Decompositions

Dr. Sepideh Hajipour

Sharif University of Technology

Electrical Engineering

Nikoo Moradi

400101934

Date: June 2, 2024

Contents

1 Non-negative Matrix Factorization

In this part, we explore the non-negative matrix factorization (NNMF) of a given matrix $\mathbf{A}$ using the Alternating Least Squares (ALS) algorithm and Multiplicative Update algorithms. The objective is to decompose matrix $\mathbf{A}$ into two non-negative matrices $\mathbf{B}$ and $\mathbf{C}$ such that $\mathbf{A} \approx \mathbf{BC}$ and the Frobenius norm of the error matrix $\mathbf{E} = \mathbf{A} - \mathbf{BC}$ is minimized. The matrix $\mathbf{A}$ is decomposed into matrices $\mathbf{B}$ and $\mathbf{C}$ by solving the optimization problem:

$$\mathbf{A} = \mathbf{BC} + \mathbf{E}, \quad \mathbf{A} \in \mathbb{R}^{n \times m}, \mathbf{B} \in \mathbb{R}^{n \times j}, \mathbf{C} \in \mathbb{R}^{j \times m}, \mathbf{E} \in \mathbb{R}^{n \times m}$$

where $\mathbf{B}$ and $\mathbf{C}$ are non-negative matrices, and j is the number of components. The initial matrices $\mathbf{B}_0$ and $\mathbf{C}_0$ are provided.

1.A ALS Algotithm

1.A.a Implementation

The ALS algorithm iteratively updates $\mathbf{B}$ and $\mathbf{C}$ by fixing one matrix and solving for the other. The process is as follows:

1. Initialize $\mathbf{B} = \mathbf{B}_0$ and $\mathbf{C} = \mathbf{C}_0$.
2. Update $\mathbf{C}$ by solving $\mathbf{C} = \max(0, (\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\mathbf{A})$.
3. Update $\mathbf{B}$ by solving $\mathbf{B} = \max(0, \mathbf{A}\mathbf{C}^T(\mathbf{C}\mathbf{C}^T)^{-1})$.
4. Compute the error matrix $\mathbf{E} = \mathbf{A} - \mathbf{BC}$.
5. Check for convergence based on the Frobenius norm of $\mathbf{E}$.

1.A.b Results for ALS Algorithm

The results from the ALS algorithm implemented in MATLAB are compared to the built-in NNMF function in MATLAB. The matrix $\mathbf{A}$, predicted matrices $\mathbf{A}_{\text{pred}}$, and errors are as follows:

1. **Original Matrix A:**

$$\mathbf{A} = \begin{pmatrix} 45119 & 63361 & 28697 & 7393 \\ 114067 & 24749 & 83233 & 44621 \\ 9677 & 36887 & 81629 & 6113 \\ 67213 & 11701 & 128879 & 79777 \\ 2017 & 102407 & 41611 & 2293 \\ 8117 & 19793 & 85121 & 1217 \end{pmatrix}$$

2. **Predicted Matrix using MATLAB ALS**

$$\mathbf{A}_{\text{pred\_matlab\_als}} = 1.0e+05 \times \begin{pmatrix} 0.4263 & 0.6429 & 0.2823 & 0.1439 \\ 1.0907 & 0.2660 & 0.7979 & 0.5792 \\ 0.0554 & 0.3842 & 0.7877 & 0.1714 \\ 0.7419 & 0.0912 & 1.3369 & 0.6120 \\ 0.0684 & 1.0184 & 0.4469 & 0 \\ 0.0202 & 0.2218 & 0.8067 & 0.1856 \end{pmatrix}$$

3. **Predicted Matrix using Custom ALS**

$$\mathbf{A}_{\text{pred\_als}} = 1.0e+05 \times \begin{pmatrix} 0.4265 & 0.6428 & 0.2822 & 0.1435 \\ 1.0912 & 0.2659 & 0.7980 & 0.5785 \\ 0.0554 & 0.3843 & 0.7876 & 0.1718 \\ 0.7417 & 0.0912 & 1.3370 & 0.6119 \\ 0.0678 & 1.0185 & 0.4477 & 0 \\ 0.0207 & 0.2219 & 0.8064 & 0.1863 \end{pmatrix}$$

4. **Error Comparison**

   - Error using MATLAB NNMF ALS: error_nnmf_als $= 3.5326e+04$
   - Error using Custom ALS: error_my_als $= 3.5321e+04$

1.B Multiplicative Algorithm

1.B.a Implementation

The Multiplicative algorithm updates $\mathbf{B}$ and $\mathbf{C}$ using element-wise multiplication and division to ensure non-negativity. The process is as follows:

1. Initialize $\mathbf{B} = \mathbf{B}_0$ and $\mathbf{C} = \mathbf{C}_0$.
2. Update $\mathbf{C}$ by $\mathbf{C} = \mathbf{C} \cdot \frac{\mathbf{B}^T\mathbf{A}}{\mathbf{B}^T\mathbf{B}\mathbf{C}+\epsilon}$.
3. Update $\mathbf{B}$ by $\mathbf{B} = \mathbf{B} \cdot \frac{\mathbf{A}\mathbf{C}^T}{\mathbf{B}\mathbf{C}\mathbf{C}^T+\epsilon}$.
4. Compute the error matrix $\mathbf{E} = \mathbf{A} - \mathbf{B}\mathbf{C}$.
5. Check for convergence based on the Frobenius norm of $\mathbf{E}$.

1.B.b Results for Multiplicative Algorithm

The results from the Multiplicative algorithm implemented in MATLAB are compared to the built-in NNMF function in MATLAB. The matrix $\mathbf{A}$, predicted matrices $\mathbf{A}_{\text{pred}}$, and errors are as follows:

1. **Predicted Matrix using MATLAB Multiplicative Algorithm**

$$\mathbf{A}_{\text{pred\_matlab\_mult}} = 1.0e+05 \times \begin{pmatrix} 0.4116 & 0.6298 & 0.3065 & 0.1556 \\ 1.0983 & 0.2614 & 0.8024 & 0.5647 \\ 0.0573 & 0.3798 & 0.7919 & 0.1659 \\ 0.7528 & 0.0960 & 1.3390 & 0.5811 \\ 0.0756 & 1.0176 & 0.4219 & 0.0171 \\ 0.0236 & 0.2149 & 0.8128 & 0.1762 \end{pmatrix}$$

2. **Predicted Matrix using Custom Multiplicative Algorithm**

$$\mathbf{A}_{\text{pred\_mult}} = 1.0e+05 \times \begin{pmatrix} 0.4116 & 0.6298 & 0.3065 & 0.1556 \\ 1.0983 & 0.2614 & 0.8024 & 0.5647 \\ 0.0573 & 0.3798 & 0.7919 & 0.1659 \\ 0.7528 & 0.0960 & 1.3390 & 0.5811 \\ 0.0756 & 1.0176 & 0.4219 & 0.0171 \\ 0.0236 & 0.2149 & 0.8128 & 0.1762 \end{pmatrix}$$

3. **Error Comparison**

   - Error using MATLAB NNMF Multiplicative: error_nnmf_mult = $3.6110e+04$
   - Error using Custom Multiplicative: error_my_mult = $3.6110e+04$

1.C SNR Analysis

To validate the correctness of the implemented algorithms, we perform the following steps:

1.C.a Generating Random Matrices

Random matrices $\mathbf{B}$ and $\mathbf{C}$ are generated from a uniform distribution.

1.C.b Defining SNR

The SNR is defined as follows:

$$SNR = 20 \log_{10} \left(\frac{\|\mathbf{BC}\|_F}{\alpha \|\mathbf{E}\|_F} \right)$$

For different SNR values (-10 dB, 0 dB, 10 dB, 30 dB, 50 dB), the scaling factor α is determined, and the matrix $\mathbf{A} = \mathbf{BC} + \alpha\mathbf{E}$ is generated.

1.C.c NMF Decomposition (j=3)

The NMF decomposition is performed using four algorithms (ALS, Multiplicative, MATLAB NNMF ALS, and MATLAB NNMF Multiplicative) and the Frobenius norm of the error is calculated. Errors using each algorithms for each SNR level are shown below:

```
E_my_als =

    2.4838    0.7444    0.2096    0.0908    0.0721


E_my_mult =

    2.4838    0.7444    0.2096    0.0225    0.0087


E_nnmf_als =

    2.4857    0.8068    0.2862    0.3697    0.2911


E_nnmf_mult =

    2.5668    0.7727    0.2304    0.0907    0.0922
```

1.C.d NMF Decomposition for different factorization ranks

The NMF decomposition is performed using four algorithms (ALS, Multiplicative, MATLAB NNMF ALS, and MATLAB NNMF Multiplicative) and the Frobenius norm of the error is calculated for different values of j (2, 3, 4) and averaged over 10 random matrices for each SNR value. The error versus SNR plots for each value of j are shown below.

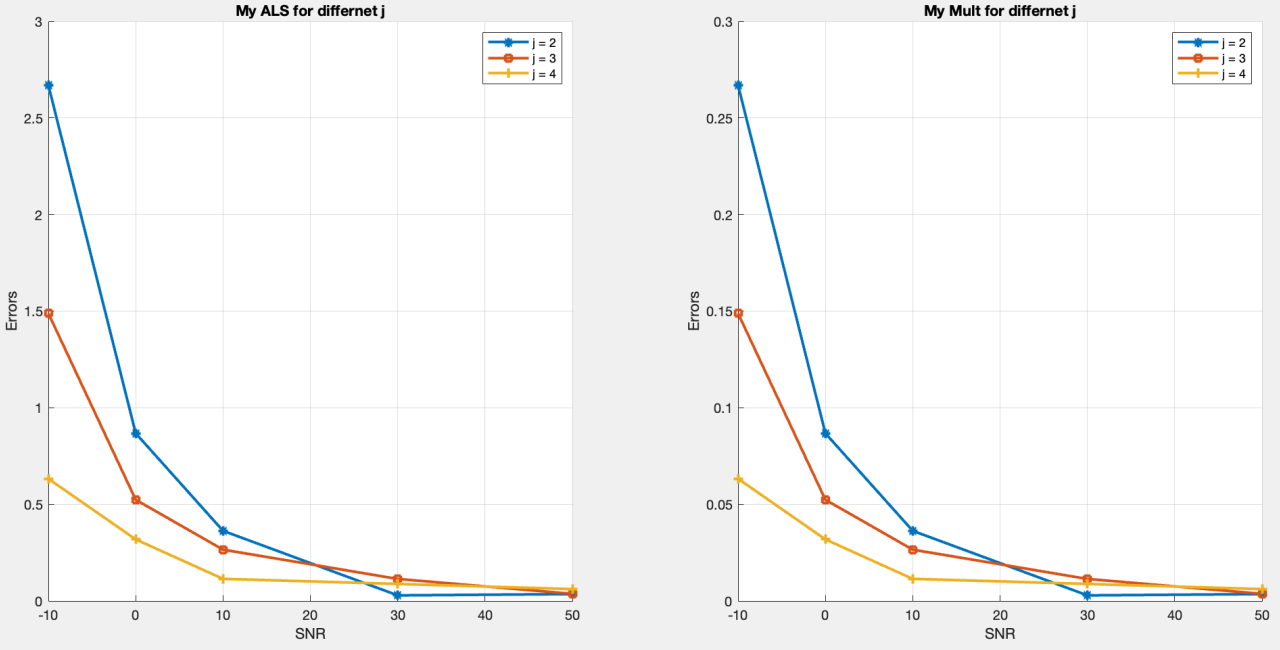

Figure 1: Error vs. SNR for different values of j using custom ALS and Multiplicative algorithms

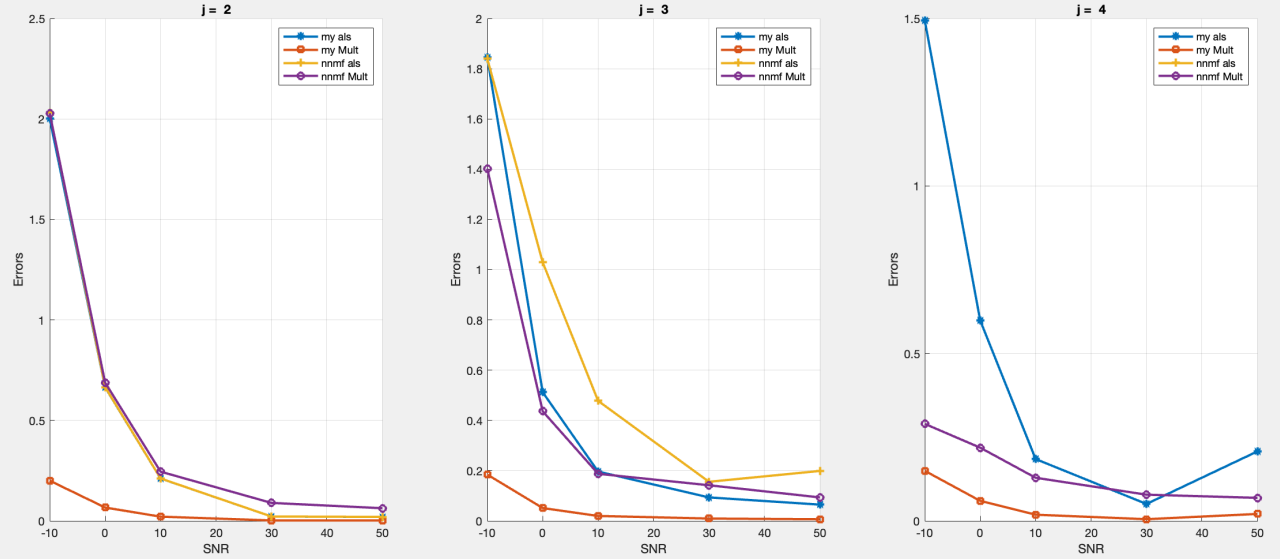

Figure 2: Error vs. SNR for different values of j comparing custom and MATLAB algorithms

1.D Results

In this assignment, we successfully implemented the ALS and Multiplicative algorithms for non-negative matrix factorization and validated their performance against MATLAB's built-in NNMF function. Both custom implementations produced results that are consistent with the built-in function, demonstrating their correctness and effectiveness. We cans see that results from both custom implementations (My ALS and My Mult) closely match those from the MATLAB built-in NNMF function. Both the predicted matrices and the error values are nearly identical, indicating that the custom implementation is accurate. The slight differences

in the predicted matrix values are due to numerical precision and rounding differences.The SNR analysis further validated the robustness of the implemented algorithms across different noise levels and matrix ranks. From 1 and **??** we can conclude that as signal to noise ratio (SNR) increases, error of all of the algorithms decreases and with more factorization rank, we get less error.

2 NMF on Swimmer Dataset

In this question, we utilize NMF to analyze a dataset consisting of 256 binary images of a swimmer. Each image represents different poses of the swimmer's arms and legs. We aim to identify and examine all possible poses by decomposing the matrix using two algorithms: the multiplicative update rule and the ALS method. Furthermore, we investigate the effect of varying the number of components (sources) on the reconstruction error for both algorithms.

2.A Methodology

2.A.a Data Preparation

The dataset consists of 256 images, each of size 9x14, which we reshape into a 256x126 matrix Y. This matrix serves as the input for the NMF algorithms.

2.A.b Algorithms

1. **Multiplicative Update Rule:**
    - This algorithm updates the factor matrices B and C by multiplicative rules to minimize the reconstruction error.
2. **Alternating Least Squares (ALS):**
    - This algorithm alternates between fixing B and solving for C, and fixing C and solving for B, using least squares optimization.

2.B Process

1. **Error Computation:**
    - For each algorithm, we compute the reconstruction error (Frobenius norm) for a range of component numbers (1 to 50).
    - The reconstruction error is defined as $E = Y - B \cdot C$, where Y is the original data matrix, and B and C are the factorized matrices.
2. **Visualization:**
    - Plot the reconstruction error against the number of components for both algorithms.
    - Visualize the original matrix Y.
    - Plot the factor matrices B and C and the error matrix E for the best number of components identified for each algorithm.

2.C Results

2.C.a Reconstruction Error

- The error plots for both algorithms are shown in Figure 3.

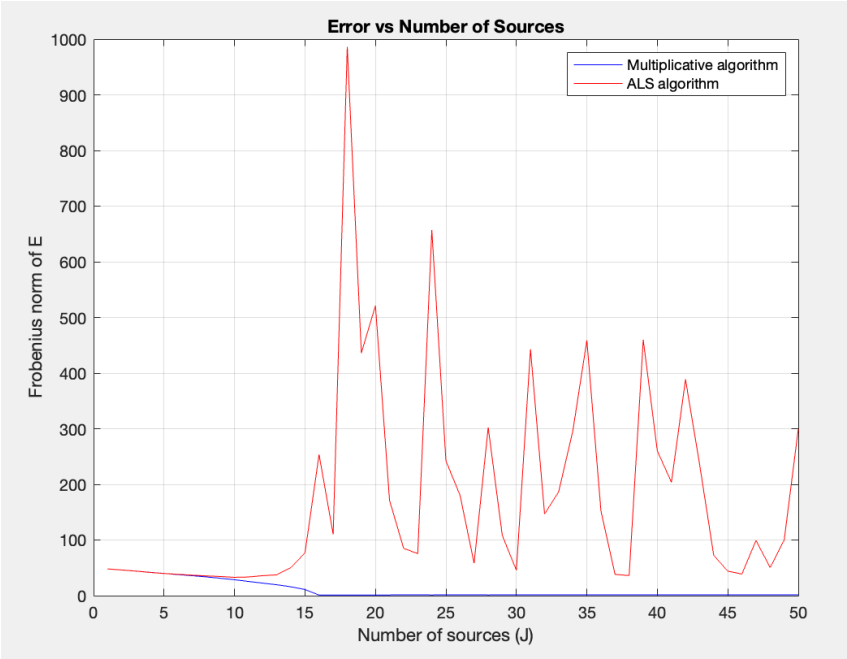

Figure 3: Error vs Number of Sources

- The multiplicative algorithm demonstrates a consistent decrease in error as the number of components increases. About 20 number of components is our optimum.
- The ALS algorithm indicates poorer performance compared to the multiplicative algorithm in case of reconstruction errors. So we choose 1 component for this algorithm.

2.D Factor Matrices

- The original data matrix Y is shown in Figure 4.

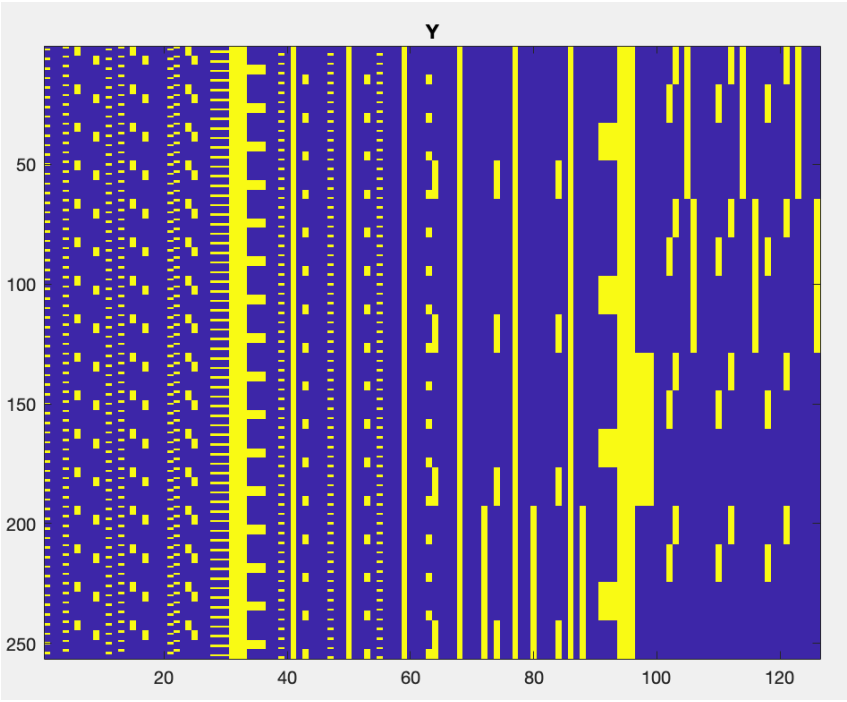

Figure 4: Y Matrix

- For the multiplicative algorithm, the factor matrix B and the components of C are shown in Figures 5 and 6, respectively.

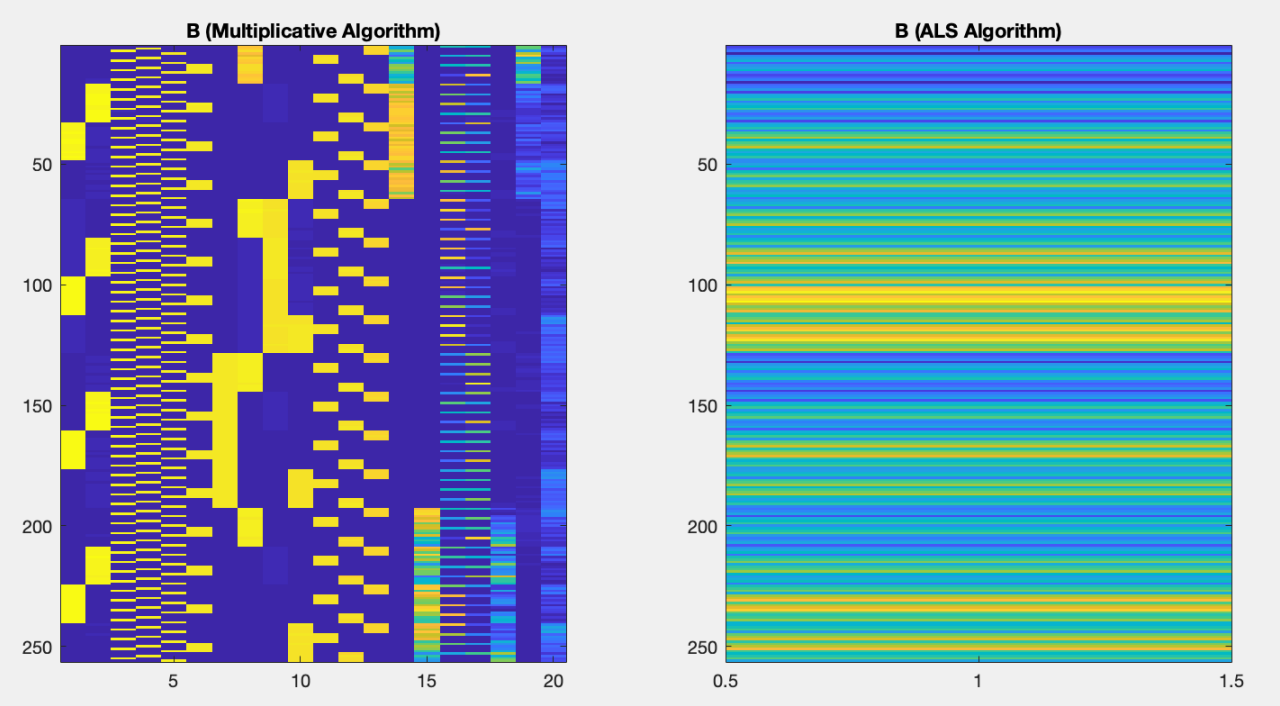

Figure 5: B Matrix

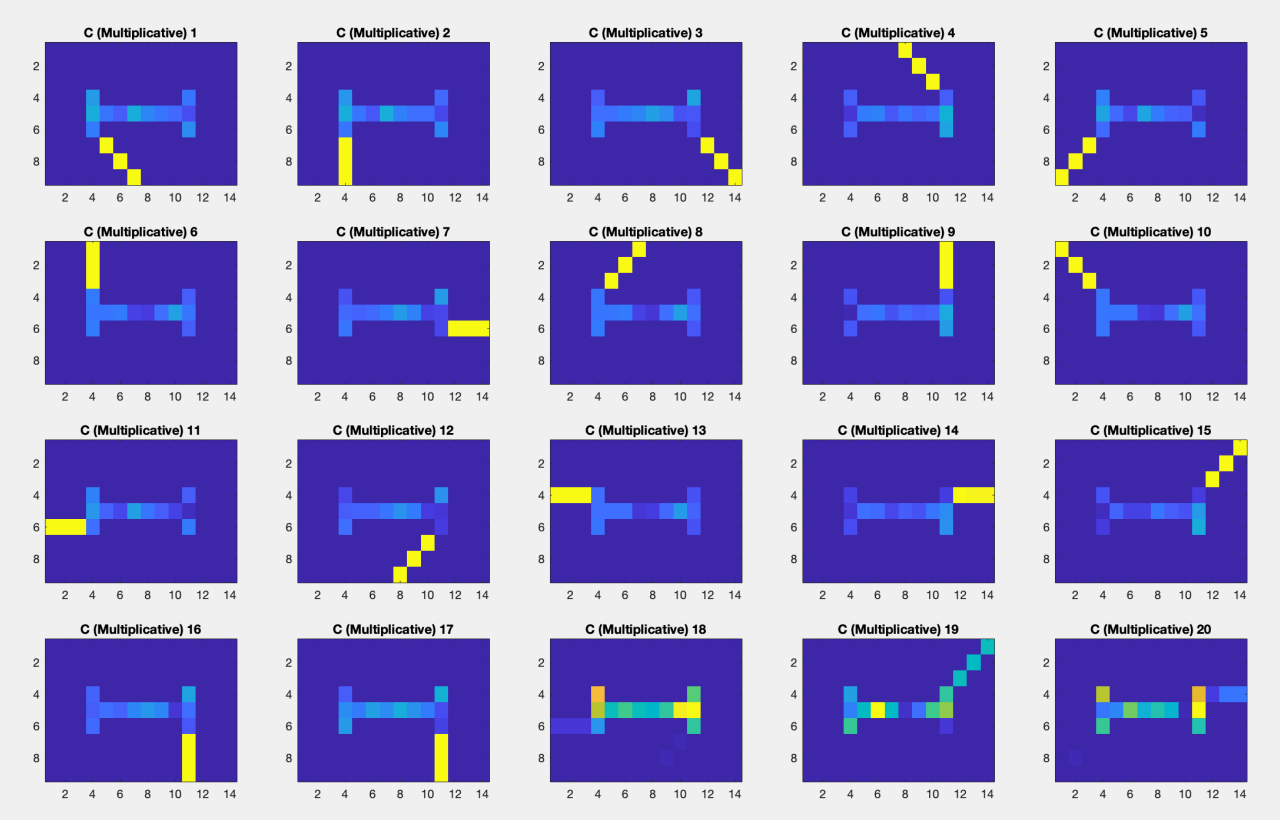

Figure 6: C Matrix Components (Multiplicative)

- For the ALS algorithm, the factor matrix B and the single component of C are shown in Figures 5 and 7. , respectively.

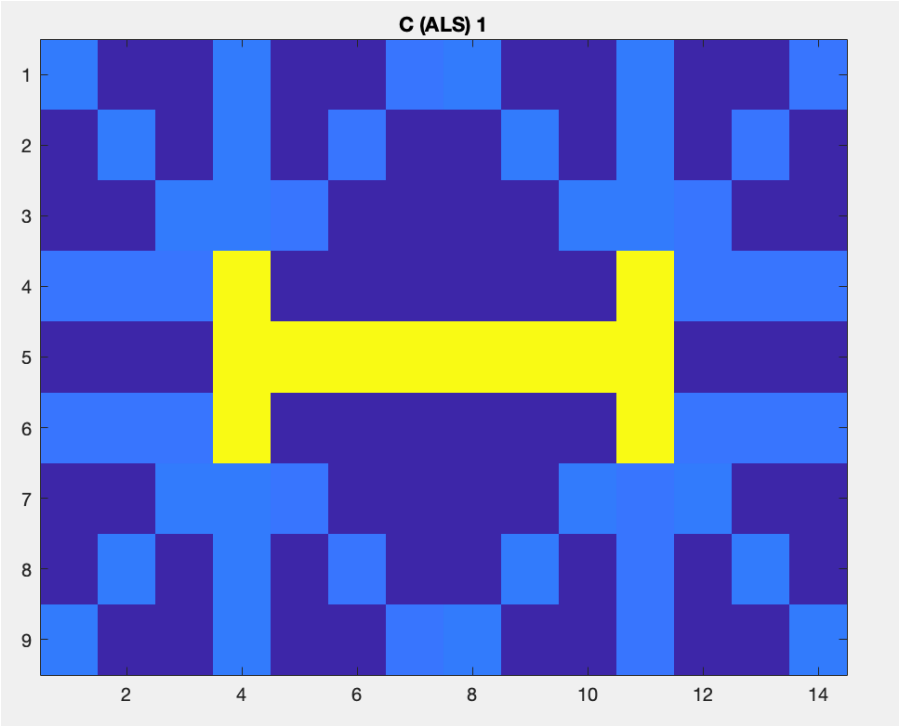

Figure 7: C Matrix Component (ALS)

2.D.a Error Matrices

- The error matrices E for both algorithms are depicted in Figure 8.

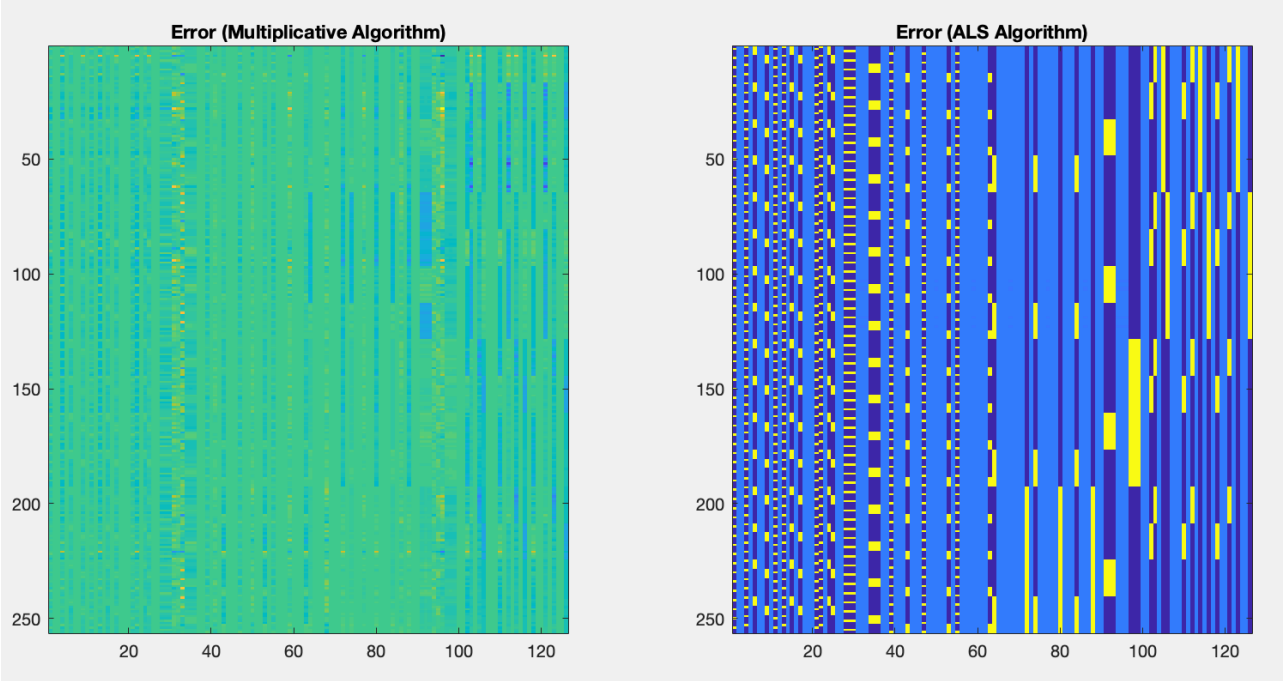

Figure 8: Error Matrices

2.E Conclusion

The multiplicative algorithm clearly outperforms the ALS algorithm in terms of reconstruction error and it is more effective for decomposing the swimmer dataset using NMF. This is evident from the error plots and the visual quality of the factor matrices. The ALS algorithm struggles with higher errors and fails to provide meaningful decompositions for higher component numbers. The optimal number of components for the multiplicative algorithm was found to be 20, while for the ALS algorithm, it was 1. This analysis highlights the importance of selecting an appropriate algorithm and the number of components for accurate matrix factorization.