Assignment 1

Tutor: Kevin Zheng, Yuhao Wu

Group members:
Zhiwen Yang, 500550636, zyan9782
Jingyi Lu, 510001238, jilu5088
Binghang Li, 530191027, bili0176
Ziyang Xiao, 530465353, zxia8013

1 Abstract

Non-negative Matrix Factorization (NMF) has become a popular technique in many domains, such as image analysis, text mining, and bioinformatics, mainly for the purpose of dimensionality reduction and feature extraction. In this report, we investigate two different variants of the NMF technique, namely L1 norm-based NMF and L2,1 norm-based NMF, on two image datasets, YaleB and ORL. In order to investigate their robustness, we added two different types of noise: block occlusion and salt-and-pepper noise. The metrics we used include Relative Reconstruction Error (RRE), Accuracy (ACC), and Normalized Mutual Information (NMI). By using these, we aim to measure reconstruction quality and clustering accuracy of the algorithms. Experimental results have shown that L1-norm NMF performs well on smaller datasets while L2,1-norm NMF exhibits superior robustness against noise, especially in larger datasets and more complex noise scenes. These results suggest that L2,1-norm NMF may outperform others in a noisy environment consistently, and the variants of NMF perform quite differently on different datasets, since the size and noise level vary.

2 Introduction

Non-negative Matrix Factorization (NMF) has been one of the essential methods in machine learning and data mining due to its special capability of dimensional reduction, data compression, and feature extraction. The main idea behind NMF is to decompose a non-negative matrix into two low-dimensional non-negative matrices. It can be mathematically expressed as $A = XY$, where the goal of NMF is to get X and Y and make their multiplication approximate to the original non-negative matrix A. This factorization is useful in applications where one wants to produce easily interpretable results. The non-negativity constraint forces the data representation to be as a combination of various parts extracted by NMF and therefore, parts-based representations are more intuitive and easier to analyze [7]. For example, in the case of face recognition, the NMF would probably separate faces into different components, such as eyes or mouth, which afterwards could be directly interpreted. In this way, the results will be more understandable and meaningful to analyze.

Moreover, NMF can be applied to various domains, such as image analysis, document clustering, and bioinformatics. By projecting data into a low-dimensional space, NMF is capable of mining some potential features, which are hard to find in high-dimensional space. For example, text mining using NMF for document clustering enables better data organization and retrieval by unveiling the latent topics in a corpus of documents. Another example is in bioinformatics, where NMF was used to analyze gene expression data, which allowed researchers to identify new biological processes and classify diseases based on genetic patterns [6]. These applications demonstrate the versatility and power of NMF in uncovering meaningful patterns or latent features from complex datasets.

The problem we try to solve is the robustness of NMF algorithms when applied to image datasets subjected to contamination by noise. Real-world data are often affected by different forms of noise or corruption, and it is quite important to check the performance of NMF algorithms under such conditions. Therefore, a performance analysis of the NMF in a noisy environment will give insight into the behavior of the method with regard to its real-world application limitations. This problem is very important in several domains dealing with data quality variations such as illumination, occlusion, or sensor noise in applications like face recognition and medical imagery.

In this study, we implemented and compared two variants of NMF: *L1-Norm Based NMF* and *L2,1-Norm Based NMF*. Through experiments, we tested the performance of these algorithms using metrics such as Relative Reconstruction Error (RRE), Clustering Accuracy (ACC), and Normalized Mutual Information (NMI). According to our findings, it suggests that *L2,1-Norm Based NMF* consistently outperforms *L1-Norm Based NMF* across both large and small datasets, particularly in handling noisy data. It proves that the *L2,1-Norm Based NMF* is more robust, especially in the case of salt-and-pepper and block occlusion noise while *L1-Norm Based NMF* can be more competitive in smaller and less noisy datasets. In general, based on these results, it can be concluded that *L2,1-Norm Based NMF* is generally more appropriate for noisy or larger-scale datasets. In contrast, *L1-Norm Based NMF* might be more suitable in simpler scenarios.

3 Related Work

NMF (non-negative matrix factorization) is a method used for high dimensional data in the realm of unsupervised learning. The main concept of it is to factorize a high-dimensional non-negative matrix into two lower-dimensional non-negative matrices, which can be expressed as:

$$A \approx XY, \tag{1}$$

where A is the non-negative matrix that needs to be factorized, X and Y are the objective matrices that optimizes its product to be as close as A[3]. Therefore, the objective function of NMF is:

$$O = \min_{X,Y} \|A - XY\|_F^2 \quad \text{s.t.} \quad X \geq 0,\ Y \geq 0 \tag{2}$$

where$| \cdot \|_F^2$ represents the Frobenius norm.

For the optimization method of NMF, we use Multiplicative Update Rules (MUR) in our project, whose rules is to make the objective function not increasing, and iteratively update X and Y to reach the local minimum solution at a stationary point of the distance when the objective function is invariant. The initial optimization process is shown below. Due to space constraints, the optimization rules will be summarized and applied in later sections.

$$\text{Fix } X, \text{ solve for } Y \quad \frac{\partial \|A - XY\|_F^2}{\partial R} = -2X^\top A + 2X^\top XY \tag{3}$$

However, the drawbacks of normal NMF is obvious. The factorization result is not unique, which may cause a local minima or non-unique solutions in some circumstances. The features of NMF are redundant, causing the sparsity to reduce and not being able to capturing the unique contibution of features. Robust NMF is introduced to solve these problems, which consider the matrix A to be represented by a clean matrix and a additive noise[1] and can be expressed as $A \approx XY + E$. Thus, the object function changes into this form:

$$O_{RobustNMF} = \|A - XY - E\|_F^2 + \lambda \sum_j \|\|E_{:,j}\|_0\|^2$$

where parameter λ controls the balance between the first and the second term, meaning that it decides the trade-off between data fitting and sparsity.

The two methods used in our article incorporates different regularization methods. The *L1-Norm Based NMF* minimize the reconstruction error with an L1 penalty on the residuals. It can detect the noise and isolate it, forming a cleaner data construction. Therefore, it is specifically suitable for dataset that contains corrupted data with additive noises[1]. But it preforms better on smaller datasets which we do not know the exact position and the type of noise, as is high precision in detection needs time to process.

The *L2,1-Norm Based NMF* utilize L1 norm regularization across rows and columns of the matrices and L2 norm regularization on individual elements [3].This approach is particularly better when data have different kinds of noises, which is a common scenario in larger datasets. Also, our experiments shows that when dealing with structured noise like salt-and-pepper issue, *L2,1-Norm Based NMF* performs better than *L1-Norm Based NMF* too.

4 Methods

4.1 Preprocessing

Pre-processing is one of the main steps in machine learning and data analysis as it helps transform the unstructured data into a form that supports efficient processing by algorithms [8]. These decisions taken at this stage affect highly the performance and results of the applied models. Some of the frequently utilized steps in data pre-processing include, but not limited to, data cleaning, normalisation, standardisation, dimensionality reduction, data partitioning, centering and augmentation. These procedures are employed in order to remove noise, minimize the time complexity, and enhance the stability of the model. In our study, we performed the following pre-processing steps on the datasets:

1. **Grayscale Conversion**:
   All of the images were converted to grayscale in order to give simplify data representation and keep the computational requirements to a minimum. In facial recognition tasks, grayscale images help in capturing minute variations of the intensity levels. Reduction of the complexity of the data in terms of this step is achieved by removing the color information that is not necessary in our case [5].
2. **Resizing**:
   - **ORL Dataset**: Images were resized by a factor of 2, resulting in dimensions of 46×56 pixels.
   - **Extended YaleB Dataset**: Images were resized by a factor of 4, resulting in dimensions of 42×48 pixels.

   Resizing reduces the dimensionality of the data, thereby decreasing computational complexity and memory requirements. By reducing the image size, we improved the training process while retaining essential facial features necessary for accurate recognition.
3. **Flattening**:
   Each two-dimensional image was converted to a one-dimensional feature vector. That transformation arranges the image data in a matrix format where each column represents a single image. Flattening is highly important before the application of matrix factorisation methods, as these methods work with matrices whose columns are data points[2].
4. **Data Utilization Without Normalization**: Although normalization is a common preprocessing step to scale data to a standard range (e.g.[0,1]), we chose to use the original pixel values ranging from 0 to 255. Since Non-negative Matrix Factorization (NMF) requires non-negative data, the original pixel values are suitable for our algorithms.

   Rationale

   - **Algorithm Requirements**: The NMF algorithms we implemented are designed to work with non-negative data, and the pixel values inherently satisfy this requirement.
   - **Avoiding Numerical Instability**: In some cases, normalization can introduce numerical instability, especially if not handled carefully.
   - **Empirical Performance**: We observed that using the original pixel values did not adversely affect the performance of our models.

Considerations

- While normalization can improve numerical stability and convergence, it is not strictly necessary in all cases.
- Future work could explore the impact of normalization on the performance of the algorithms.

5. **Noise Addition (for Evaluation)**:

   To evaluate the robustness of the algorithms, we introduced noise into the datasets. We added **block occlusion noise** to simulate real-world scenarios where parts of the face may be obstructed.

   Block Occlusion Noise

   - **Description**: Block occlusion simulates situations where parts of the face are covered by objects like sunglasses, scarves, or hands. A square block of pixels is overlaid on a random location in each image, effectively hiding part of the facial features.
   - **Parameters**:
     - **Block Size** (b): 12 pixels (i.e., a 12×12 pixel square).
     - **Intensity Value**: The pixel values within the block are set to 255 (maximum intensity for 8-bit grayscale images), creating a white square.
   - **Implementation**:
     - For each image X_{original}, we created a noise mask N where $N_{ij} = 255$ within the block and $N_{ij} = 0$ elsewhere.
     - The noisy image X_{noisy} is obtained by adding the noise mask to the original image:

$$X_{\text{noisy}} = X_{\text{original}} + N \tag{4}$$

   - **Purpose**:
     - The occlusion challenges the algorithms to reconstruct and interpret images with missing or corrupted information.
     - It allows us to assess the robustness of the NMF algorithms under conditions that mimic practical obstacles in face recognition tasks.

4.2 L1 Norm

The L1-Norm-Based Non-negative Matrix Factorization is an extension to the standard NMF, giving it robustness in noise and outlier data. Therefore, by implementing the L1-norm on the cost function, this method reduces the effect of large deviation caused due to corruption or occlusion in the data. Hence, it best suits those image datasets that have inherent noise.[4].

4.2.1 Cost Function

The cost function for L1 Norm NMF can be expressed as:

$$J(X, Y) = \|A - XY\|_1 + \lambda \|Y\|_1, \tag{5}$$

where A is the original non-negative matrix, X and Y are the non-negative matrices to be learned, $\|\cdot\|_1$ denotes the L1 norm, and λ is a regularization parameter that controls the sparsity of Y. The first term measures the reconstruction error, while the second term encourages sparsity in the factor matrix Y.

4.2.2 Optimization

To optimize the L1 Norm cost function, we utilize the **Multiplicative Update Rules** (MUR), which iteratively update the matrices X and Y. The update rules for X and Y can be derived as follows:

For updating X:

$$X \leftarrow X \odot \frac{(AY^T)}{(XYY^T + \epsilon)}, \tag{6}$$

For updating Y:

$$Y \leftarrow Y \odot \frac{(X^T A)}{(X^T XY + \lambda + \epsilon)}, \tag{7}$$

where $\odot$ denotes element-wise multiplication, and ϵ is a small positive constant added to prevent division by zero. The algorithm iterates until convergence, which is typically assessed by checking the relative change in the cost function or the factor matrices.

4.3 L2-1 Norm

The L2,1 Norm is a matrix decomposition method designed to promote sparsity and robustness, particularly suitable for handling image data that may be contaminated with noise. Unlike standard L1 Norm and L2 Norm, the L2,1 Norm applies L2 regularization at the row level and L1 regularization at the column level. This combination allows for emphasizing the contribution of each feature across the entire sample while ensuring the model's robustness against noise.

4.3.1 Cost Function

To define the cost function for the L2,1 Norm, we can extend the standard objective function of Non-negative Matrix Factorization (NMF). The cost function for L2,1 Norm can be expressed as:

$$\min_{X,Y} ||A - XY||_F^2 + \lambda ||X||_{2,1} \tag{8}$$

where:

- $||A - XY||_F^2$ is the Frobenius norm that measures the error between the original matrix A and the reconstructed matrix XY.
- λ is the regularization parameter that controls the model's complexity and balances the trade-off between reconstruction error and sparsity.
- $||X||_{2,1} = \sum_{j=1}^{m} ||X_j||_2$ is the L2,1 Norm, which encourages row sparsity by calculating the L2 norm for each row and summing them up.

This cost function encourages the extraction of significant features from the data while suppressing the influence of noise on the reconstruction results, thus achieving a more robust decomposition.

4.3.2 Optimization

To optimize the L2,1 Norm cost function, we can employ the Alternating Minimization Algorithm. This algorithm iteratively optimizes the two matrices X and Y to minimize the cost function.

1. **Initialization**: Randomly initialize the matrices X and Y such that both X and Y are non-negative.
2. **Alternating Optimization**:
   - **Optimize X while fixing Y**:
     $$X = \arg\min_X ||A - XY||_F^2 + \lambda ||X||_{2,1} \tag{9}$$
     This step can be accomplished using gradient descent methods or other optimization techniques that incorporate L2,1 regularization.
   - **Optimize Y while fixing X**:
     $$Y = \arg\min_Y ||A - XY||_F^2 \tag{10}$$
     This step can be optimized using standard least squares methods.

3. **Iteration**: Repeat the above steps until the cost function converges or a predetermined number of iterations is reached.

This optimization method leverages the characteristics of the L2,1 Norm, allowing the reconstruction matrix to exhibit improved robustness in the presence of noise, thereby enhancing the performance of the algorithm in practical applications.

4.4 Noise

In this study, we introduce two types of noise: block-occlusion noise and salt and pepper noise, to evaluate the robustness of the proposed methods.

4.4.1 Block-occlusion noise

Block occlusion represents scenarios where parts of the face are hidden due to obstructions such as sunglasses, hands, or scarves. To simulate this, we create a noise mask that occludes a block of pixels in the original image.

The parameters for block-occlusion noise include:

- **Block Size** (b): The size of the occluding block is set to 12 pixels, resulting in a 12×12 pixel square.
- **Intensity Value**: The pixel values within the block are set to 255, representing maximum intensity for 8-bit grayscale images, creating a white square overlay.

The implementation involves creating a noise mask N where $N_{ij} = 255$ within the block and $N_{ij} = 0$ elsewhere. The noisy image X_{noisy} is obtained by adding the noise mask to the original image as follows:

$$X_{\text{noisy}} = X_{\text{original}} + N \tag{11}$$

This type of occlusion challenges the algorithms to accurately reconstruct and interpret images with incomplete information, thereby assessing the robustness of the NMF algorithms in practical face recognition scenarios.

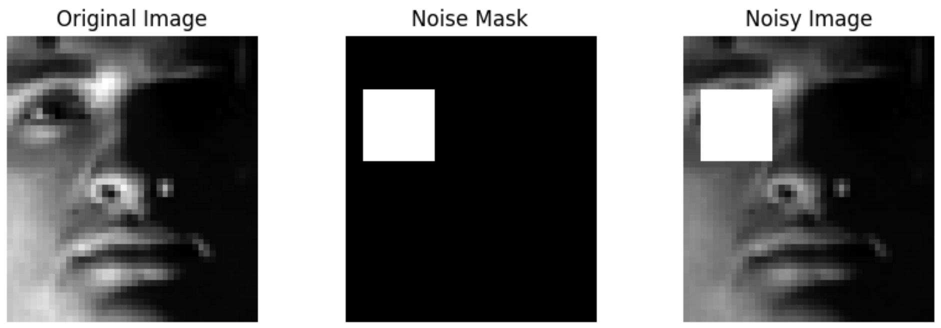

Figure 1: YaleB Dataset Block Noise

4.4.2 Salt and pepper noise

Salt and pepper noise is a type of random noise that presents itself as sparsely occurring white and black pixels in an image. This noise mimics the effects of transmission errors or sensor failures during image acquisition.

The parameters for salt and pepper noise include:

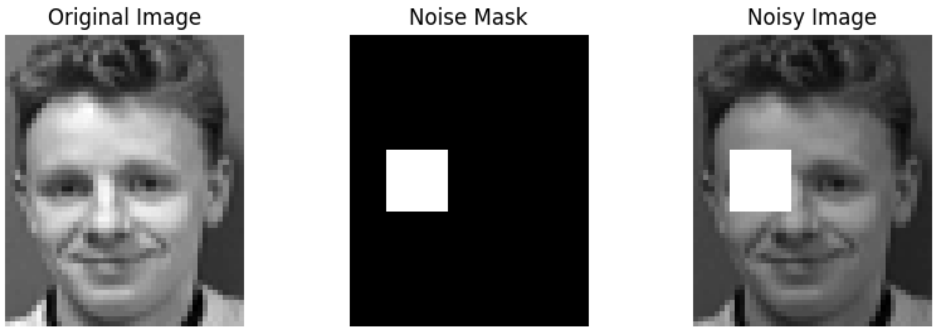

Figure 2: ORL Dataset Block Noise

- **Noise Density (p)**: This parameter defines the proportion of pixels in the image that will be altered to either maximum intensity (255) or minimum intensity (0). A typical value for p might be set to 0.02, meaning 2% of the total pixels will be affected.

The implementation involves iterating through each pixel of the original image X_{original} and randomly assigning the pixel to 0 or 255 with a probability of p for salt and p for pepper, respectively. The resulting noisy image X_{noisy} is defined as:

$$X_{\text{noisy}} = \begin{cases} 255 & \text{with probability } \frac{p}{2} \\ 0 & \text{with probability } \frac{p}{2} \\ X_{\text{original}} & \text{otherwise} \end{cases} \tag{12}$$

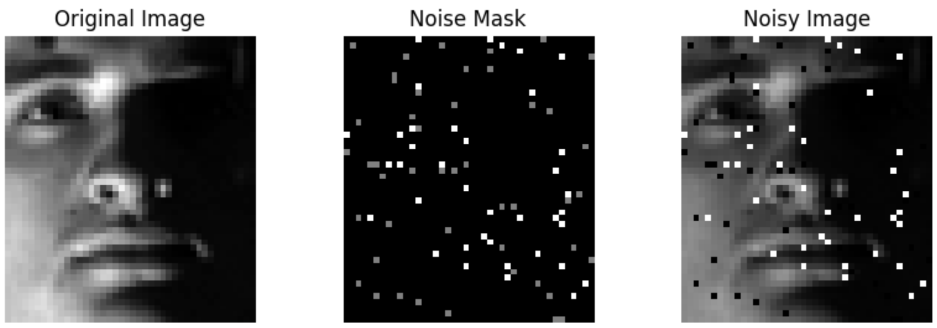

Figure 3: YaleB Dataset Salt-and-Pepper Noise

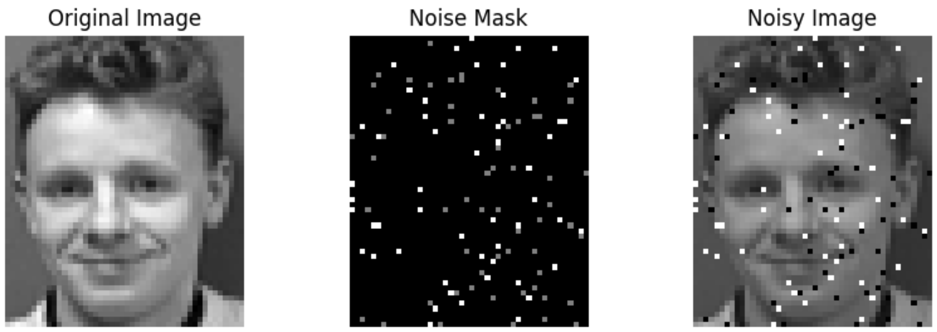

Figure 4: ORL Dataset Salt-and-Pepper Noise

5 Experiment

5.1 Evaluation Metrics

This section introduces the main evaluation metrics used in this study to assess the performance and effectiveness of the model results.

5.1.1 Normalized Mutual Information (NMI)

Normalized Mutual Information (NMI) is a measure used to evaluate the consistency between clustering results and true labels. It quantifies the effectiveness of clustering by measuring the mutual information between the clustering results and the true labels. The value of NMI ranges from 0 to 1, where 0 indicates no agreement and 1 indicates perfect agreement. The formula for calculating NMI is:

$$NMI(U,V) = \frac{2 \times I(U;V)}{H(U) + H(V)} \tag{13}$$

where $I(U;V)$ is the mutual information between U and V, and $H(U)$ and $H(V)$ are the entropies of U and V, respectively. NMI is an important metric for evaluating clustering quality, especially in cases where class labels are imbalanced or the number of clusters is unknown.

5.1.2 Average Accuracy

Average Accuracy is a commonly used evaluation metric for classification, measuring the model's classification performance across all classes. It is calculated by averaging the accuracy of each class. The formula for Average Accuracy is:

$$\text{Average Accuracy} = \frac{1}{C} \sum_{i=1}^{C} \frac{TP_i}{TP_i + FP_i + FN_i} \tag{14}$$

where C is the number of classes, and TP_i, FP_i, and FN_i represent the true positives, false positives, and false negatives for class i, respectively. Average Accuracy provides a comprehensive view of the model's performance across different classes and is particularly useful in multi-class classification scenarios.

5.1.3 Relative Reconstruction Error (RRE)

Relative Reconstruction Error (RRE) is an evaluation metric used to assess the quality of reconstruction in dimensionality reduction or matrix factorization models. It quantifies the difference between the original data and the reconstructed data, providing a measure of how well the model retains the structure of the original data. A lower RRE indicates better reconstruction quality and a more effective model.

The formula for RRE is:

$$RRE = \frac{\|X - \hat{X}\|_F}{\|X\|_F} \tag{15}$$

where X represents the original data matrix, $\hat{X}$ is the reconstructed data matrix, and $\|\cdot\|_F$ denotes the Frobenius norm. The Frobenius norm measures the overall difference between two matrices, and RRE reflects the relative error in reconstruction compared to the original matrix.

In this study, RRE is used to evaluate the effectiveness of the model in tasks involving dimensionality reduction and reconstruction, especially when dealing with noisy or corrupted data. A lower RRE implies that the model has successfully preserved important information from the original data.

5.2 Results

This section evaluates the performance of L1-Norm NMF and L2,1-Norm NMF algorithms on the YaleB and ORL datasets with two types of noise: block noise and salt-and-pepper noise. The parameter K refers to the number of clusters or components used in the NMF decomposition. We report three performance metrics:

- RRE (Relative Reconstruction Error): Measures the quality of reconstruction.
- ACC (Clustering Accuracy): Indicates the accuracy of the clustering results.
- NMI (Normalized Mutual Information): Reflects the quality of clustering compared to ground truth.

The tables below summarize the performance of the two methods under different noise conditions.

Table 1: Performance on YaleB Dataset with Block Noise

Metrics / Methods	K=10	K=30	K=50
RRE			
L1-Norm NMF	0.52457	0.52066	0.51796
L2,1-Norm NMF	0.39384	0.29005	0.25784
ACC			
L1-Norm NMF	0.08748	0.08886	0.08840
L2,1-Norm NMF	0.09006	0.09890	0.10562
NMI			
L1-Norm NMF	0.11664	0.11799	0.11780
L2,1-Norm NMF	0.11694	0.13752	0.15074

Discussion: As shown in Table 1, the L2,1-Norm NMF method consistently outperforms L1-Norm NMF in all metrics. It achieves lower RRE, indicating better reconstruction quality, and higher ACC and NMI values, suggesting superior clustering performance.

Table 2: Performance on YaleB Dataset with Salt-and-Pepper Noise

Metrics / Methods	K=10	K=30	K=50
RRE			
L1-Norm NMF	0.44073	0.43458	0.43172
L2,1-Norm NMF	0.38333	0.35667	0.34549
ACC			
L1-Norm NMF	0.08646	0.08619	0.08665
L2,1-Norm NMF	0.13785	0.21611	0.23950
NMI			
L1-Norm NMF	0.11121	0.11476	0.11290
L2,1-Norm NMF	0.21182	0.32672	0.35411

Discussion: In Table 2, similar trends are observed under salt-and-pepper noise conditions. L2,1-Norm NMF again demonstrates robustness, with lower RRE and significantly better ACC and NMI as K increases, compared to L1-Norm NMF.

Table 3: Performance on ORL Dataset with Block Noise

Metrics / Methods	K=10	K=30	K=50
RRE			
L1-Norm NMF	0.35558	0.35160	0.34915
L2,1-Norm NMF	0.25628	0.19912	0.17538
ACC			
L1-Norm NMF	0.07412	0.07255	0.07203
L2,1-Norm NMF	0.09240	0.10627	0.11175
NMI			
L1-Norm NMF	0.10512	0.10814	0.10677
L2,1-Norm NMF	0.12799	0.15877	0.17716

Discussion: Table 3 shows the performance on the ORL dataset with block noise. L2,1-Norm NMF outperforms L1-Norm NMF across all K values, providing significantly better clustering accuracy (ACC) and NMI, indicating superior feature extraction in the presence of block noise.

Table 4: Performance on ORL Dataset with Salt-and-Pepper Noise

Metrics / Methods	K=10	K=30	K=50
RRE			
L1-Norm NMF	0.40575	0.40291	0.40071
L2,1-Norm NMF	0.32022	0.29735	0.28710
ACC			
L1-Norm NMF	0.08357	0.08409	0.08373
L2,1-Norm NMF	0.12185	0.15911	0.16474
NMI			
L1-Norm NMF	0.11218	0.11554	0.11492
L2,1-Norm NMF	0.17325	0.23317	0.24510

Discussion: Finally, Table 4 presents the results on ORL with salt-and-pepper noise. Similar to previous datasets, L2,1-Norm NMF outperforms L1-Norm NMF across all metrics, particularly in ACC and NMI, indicating its ability to handle noisy data effectively.

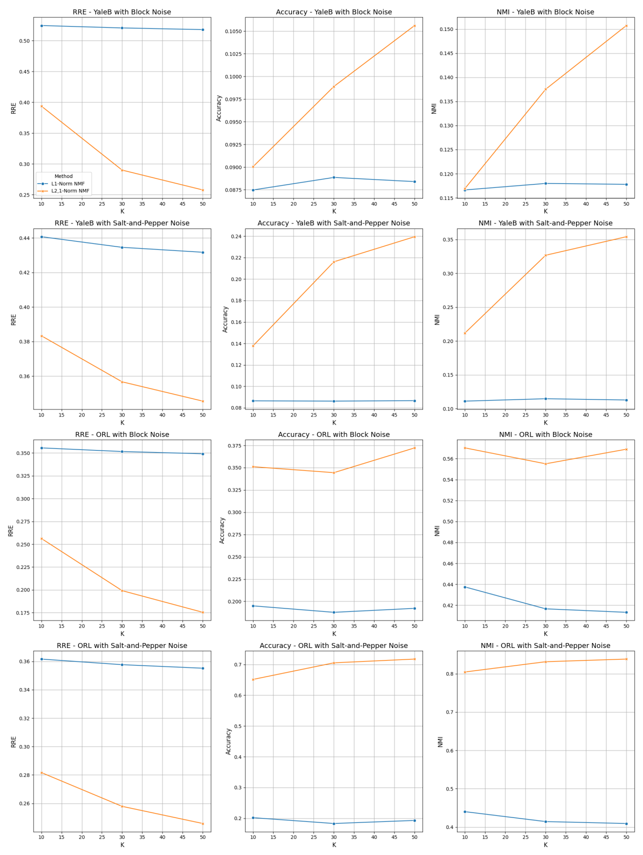

Figure 5: Performance metrics for the YaleB and ORL datasets with different noise types.

Figure 5 visualizes the performance of the L1-Norm NMF and L2,1-Norm NMF methods across different datasets and noise types. The metrics presented include RRE, Accuracy, and NMI for each method across various values of K. These visualizations offer an intuitive comparison between the methods and help in identifying trends across the noise types and datasets.

6 Conclusion

In this paper, we will contrast two algorithms of NMF: **L1-Norm NMF** and **L2,1-Norm NMF**, applied to the YaleB and ORL face image datasets, considering two types of noises, specifically *block occlusion noise* and *salt-and-pepper noise*. The objective of this work is to study the robustness of these algorithms for image reconstruction and clustering when images are corrupted by noise, which is an important factor in many practical applications related, for example, to face recognition, where the quality of the data may be quite diverse.

6.1 Summary of Findings

1. **Consistent Superiority of L2,1-Norm NMF:**

    - Across two given datasets, noise types, and values of K (the number of components), the **L2,1-Norm NMF** algorithm consistently outperformed the **L1-Norm NMF**.
    - *Reconstruction Quality (RRE):* L2,1-Norm NMF achieved lower Relative Reconstruction Error (RRE) in all scenarios, indicating superior reconstruction capabilities. For example, on the YaleB dataset with the block noise, RRE for L2,1-Norm NMF decreased from 0.39384 to 0.25784 as K increased from 10 to 50, significantly lower than the L1-Norm NMF's RRE.

- *Clustering Performance (ACC and NMI):* L2,1-Norm NMF had higher Average Accuracy (ACC) and Normalized Mutual Information (NMI) than L1-Norm NMF, which showed better clustering performance. Particularly, on the YaleB dataset with salt-and-pepper noise and $K = 50$, L2,1-Norm NMF achieved an ACC of 0.23950 and NMI of 0.35411, compared to 0.08665 and 0.11290, respectively, for L1-Norm NMF.

2. **Impact of Noise Types**:
   - *Block Noise vs. Salt-and-Pepper Noise:* While both noise types adversely affected performance, the degradation was more pronounced under salt-and-pepper noise. However, L2,1-Norm NMF showed greater resilience, and maintained better reconstruction and clustering metrics compared to L1-Norm NMF.
   - *Scalability with K:* As shown in Figure 5, increasing the number of components K generally led to better performance for both algorithms, but L2,1-Norm NMF has a more significant advantage, especially in higher dimensions.
3. **Low Clustering Accuracies:**
   - Even though L2,1-Norm NMF performed quite better, the overall clustering accuracy for both approaches was still comparatively low. This reflects some of the intrinsic difficulties of clustering in high-dimensional noisy image data and suggests further improvement might be possible.

6.2 Future Work

1. **Data Normalization:**
   - Implementing normalization techniques to scale pixel values to a standard range (e.g., [0, 1]) could enhance numerical stability and potentially improve convergence rates and the overall performance.
2. **Algorithm Enhancements:**
   - Exploring additional NMF variants or integrating regularization techniques could further improve robustness and clustering accuracy.
   - Incorporating sparsity constraints or adaptive weighting schemes might help in capturing more discriminative features.
3. **Extended Noise Models:**
   - Evaluating the algorithms under more different noise conditions, such as Gaussian noise or real-world occlusions, would provide a more comprehensive understanding of their robustness.
   - Investigating the impact of varying noise levels and patterns could help in developing noise-aware algorithms.
4. **Scalability and Real-World Applications:**
   - Testing the algorithms on larger, more diverse datasets would assess their scalability and practical applicability.
   - Applying the methods to real-world tasks, such as surveillance or medical imaging, could validate their effectiveness in operational settings.
5. **Clustering Techniques:**
   - Combining NMF with more advanced clustering algorithms or incorporating supervised learning elements might enhance clustering performance.
   - Exploring subspace clustering or manifold learning methods in conjunction with NMF could yield better results.

7 Appendix

7.1 How to Run the Code

1. Access the code and datasets from the provided Google Cloud folder at: Google Drive Link.

2. Open the `COMP5328_A1_Group98.ipynb` notebook file in Google Colab.
3. Upload the dataset files including ORL and YaleB datasets to the `/content/` directory in Colab. If you are using a different directory, adjust the file paths in the code accordingly.
4. Ensure all files are fully uploaded before running the code.
5. Ensure that all necessary libraries are installed in the Colab environment, such as NumPy, Pandas, Matplotlib, and Scikit-learn.
6. Execute all cells in the notebook by pressing `Ctrl+F9` or selecting `Runtime -> Run all` from the Colab menu.

7.2 Group Contribution

- **Zhiwen Yang & Jingyi Lu**: Responsible for writing the code for L1 norm based NMF and contributing to the corresponding sections of the report.
- **Binghang Li & Ziyang Xiao**: Responsible for writing the code for L2,1 norm based NMF and contributing to the corresponding sections of the report.

References

[1] Rongrong Ji Baodi Liu Bin Shen, Luo Si. Robust nonnegative matrix factorization via l1 norm regularization. *arXiv preprint*, 1204.2311:1–22, 2012.

[2] P. T. Boufounos, H. Mansour, S. Rane, and A. Vetro. Dimensionality reduction of visual features for efficient retrieval and classification. *APSIPA Transactions on Signal and Information Processing*, 5(1), 2016.

[3] Yixuan Xu Cheng Zeng, Jiaqi Tian. Analyze the robustness of three nmf algorithms (robust nmf with l1 norm, l2-1 norm nmf, l2 nmf). *arXiv preprint*, 2312.01357:1–22, 2023.

[4] A. Díaz and David Steele. Analysis of the robustness of nmf algorithms. *arXiv.Org*, 2021.

[5] Christopher Kanan and Garrison W. Cottrell. Color-to-grayscale: Does the method matter in image recognition? *PloS One*, 7(1):e29740–e29740, 2012.

[6] Hyunsoo Kim and Haesun Park. Sparse non-negative matrix factorization for clustering. *Journal of Machine Learning Research*, 12:153–189, 2011.

[7] Nuno Lopes and Bernardete Ribeiro. *Non-Negative Matrix Factorization (NMF)*, volume 7 of *Studies in Big Data*. Springer, 2015.

[8] N. Pandey, P. K. Patnaik, and S. Gupta. Data pre-processing for machine learning models using python libraries. *International Journal of Engineering and Advanced Technology*, 9(4):1995–1999, 2020.