Analysis of the robustness of NMF algorithms

Yinghua Zhou
The University of Sydney

Abstract

Non-negative Matrix Factorization (NMF) is a commonly used technique for many practical applications including image reconstruction. While the data in practice is often subject to various types of contamination, different variants of NMF tend to exhibit divergent performance characteristics. In this study, we examine three specific NMF algorithms, namely the standard L_2-norm NMF [1], L_1-norm regularized NMF [2], $L_{2,1}$-norm NMF [3]. Our experiments include the assessments of the robustness of each NMF algorithm against three simulated noise types. This paper firstly discusses the characteristics of each NMF algorithm and the noises used, we then present the experimental results under both non-contamination and the specified noise settings, based on the discussions of the experimental setups. Results uncover several insights of the NMF algorithms, while future directions are suggested at the end.

1 Introduction

Non-negative Matrix Factorization (NMF) is a matrix factorization technique that aims to effectively approximate the original data matrix with two factorized matrices in lower dimensionality spaces, that preserve the non-negative property for all entries of both sub-matrices. It has been widely adopted in many practical applications such as gene expression analysis, text mining and image processing. Its non-negativity that effectively turns matrix factors into parts-based representation is inherently suitable for many types of data including image pixel intensity or word count.

A significant challenge of NMF algorithms in practice is their sensitivity to noise, since real-world data is often contaminated in various kinds. Therefore understanding how specific NMF algorithms perform under distinct noise settings is crucial for their reliable deployment in practice. In this study, we aim to examine three specific NMF algorithms, namely the standard L_2-norm NMF[1], L_1-norm regularized NMF [2], $L_{2,1}$-norm NMF [3]. We base our experiments on the ORL [4] and Extended YaleB [5] datasets, and assess the three NMF algorithms with the data under both uncontaminated conditions and also in the presence of three simulated noise types, namely the Gaussian noise, Salt & Pepper noise and Block Occlusion noise. We evaluate the performance of each NMF algorithm according to the Relative Reconstruction Error (RRE), Averaged Accuracy (ACC) and normalized Mutual Information (NMI) metrics, while the latter two are calculated based on K-means clustering with majority voting for each cluster. Our goal and contribution is to gain insights into the actual practical behaviors of each proposed NMF algorithm under specific contamination settings, shedding light on their relevant robustness and reliability.

2 Related work

The general idea of Non-negative Matrix Factorization (NMF) is to decompose a data matrix into two sub-matrices that can closely approximate to original one. The concept can be formally depicted as follows: Given a non-negative matrix $X \in \mathbb{R}^{m \times n}$, while each column of X represents a data sample, and thus each row represents a feature (e.g., in the context of image reconstruction, each row represents a pixel of the image, in a flattened one-dimensional shape), the NMF algorithm tries

to learn two non-negative matrix factors W and H, where $W \in \mathbb{R}^{m \times k}$ and $H \in \mathbb{R}^{k \times n}$, with k as a hyper-parameter usually chosen to be smaller than m or n, such that:

$$X \approx WH \tag{1}$$

There are several NMF variants based on this concept, while many of them tend to exhibit different characteristics. In this paper we mainly focus on the standard L_2-norm NMF[1], L_1-norm regularized NMF [2], $L_{2,1}$-norm NMF [3].

2.1 Standard L_2-norm based NMF

The standard L_2-norm NMF [1] incorporates the squared Frobenius norm to minimize the reconstruction distance that is between X and WH. In terms of optimization, the widely adopted gradient descent methods are discovered to either have slow convergence or is complicated to implement [1], while the choice of learning rate is often hard to determine. Therefore a novel proven optimization scheme of what is known as Multiplicative Update Rules (MUR) are used, that is typically faster without compromising the ease of implementation, and is specifically designed for this algorithm. The standard L_2-norm NMF typically works well when the data is not contaminated, or under the context of Gaussian noise [6], but is susceptible to outliers or noises like large partial corruptions [6, 2], where its robustness is significantly downgraded in that it tries to fit onto the noisy data.

2.2 L_1-norm regularized robust NMF

The L_1-norm regularized NMF [2] assumes the presence of large sparse additive noise by incorporating an additional noise matrix into its cost function on top of the standard L_2-norm NMF, that serves as a preserved space for predicting the positions of the potential noise, that is isolated from the reconstructed matrix to approximate noise-less reconstruction. Additionally, the noise matrix is regularized using weighted L_1-norm to control its sparsity, where the weight is set as a hyper-parameter. A specific proven optimization scheme for the relevant parameters are also proposed to corporate with the changes in the loss. This NMF algorithm is designed particularly for the scenarios where partial corruptions exist based on its assumptions, despite its effectiveness has been proven to some extent through the experiments in the corresponding paper, its robustness against different kinds of noise, under different settings are relatively less reliable without further assessment, while the additional complexity of its Multiplicative Update Rules could contribute to its higher cost of computation compared to some other NMF algorithms, and the weight for the L_1-norm on the noise matrix is often non-trivial to determine in practice.

2.3 $L_{2,1}$-norm based NMF

The $L_{2,1}$-norm NMF [3] differs from the standard L_2-norm NMF by its non-squaring property. The fact that the standard NMF squares the residual between reconstruction and target matrix under data contamination makes it susceptible to outliers and large noise, which could in theory significantly dominate the actual distances toward the clean original data matrix. The $L_{2,1}$-norm NMF with the aim to resolve such an issue by proposing its non-squaring property that adds a square root to the inner summation, that mitigates the dominance of the impact of outliers. The $L_{2,1}$-norm NMF has been demonstrated comparative performance at the presence of outliers and also clustering results across various datasets, while it is said to converge with smaller number of iterations compared to the standard one. However, its effectiveness of reconstruction and its generalizability to different kinds of noise under different settings are yet to be revealed.

3 Methods

In this section, we will present the details of each NMF algorithm that we will be analyzing and the noises we will be using in our experiments, while giving an overview of the respective robustness of each NMF algorithm from the theoretical point of view.

3.1 Pre-processing

We employ image size reduction on the experimented datasets for better computational efficiency and potentially mitigating the impact of irrelevant features. We also apply normalization to turn the image pixels into the 0-1 scale, which is a commonly adopted strategy that potentially prompt stability and better training dynamics, while being more natural to the algorithms and throughout the implementation.

3.2 Standard L_2-norm based NMF

The standard L_2-norm NMF[1] aims to minimize the squared Frobenius norm distance between the original data matrix and the reconstructed data matrix by two decomposed non-negative sub-matrices.

3.2.1 Cost Function

The cost function of the standard L_2-norm NMF can be defined as follows, where $||\cdot||_F$ denotes the Frobenius norm, and $X \in \mathbb{R}^{m\times n}$, $W \in \mathbb{R}^{m\times k}$, $H \in \mathbb{R}^{k\times n}$:

$$||X - WH||_F^2 = \sum_{i,j} \left(X_{ij} - (WH)_{ij}\right)^2 \tag{2}$$

which we attempt to minimize, in regards to W and H, with subject to the constraints of $W \geq 0$ and $H \geq 0$.

3.2.2 Optimization

Despite the simplicity of its cost function, it is actually convex with respect to either W only or H only, but not to both. Therefore optimization in practice involves updating one of them respectively at a time, that makes finding global minima unrealistic. However finding local minima is often sufficient, and can be achieved by using several numerical optimization techniques.

Conventional additive gradient descent based methods either suffer from slow convergence or complex implementation. Therefore the dedicated Multiplicative Update Rules are proposed [1], as follows:

$$W_{ij} \leftarrow W_{ij} \frac{(XH^T)_{ij}}{(WHH^T)_{ij}} \tag{3}$$

$$H_{ij} \leftarrow H_{ij} \frac{(W^T X)_{ij}}{(W^T WH)_{ij}} \tag{4}$$

This scheme of updating is derived from the conventional gradient descent, that has been convergence-proven, while it typically gives a good balance between convergence speed and ease of implementation.

As a side note, it is actually straightforward to notice that if the reconstruction WH perfectly approximates the original data matrix X, the corresponding WH terms in the denominator of both update rules can be treated as X, effectively making the updated W and H stay still, that is, the convergence is reached.

3.2.3 Characteristics

The standard L_2-norm NMF [1] generally performs well when the data is either free from contamination or affected only by Gaussian noise. This is because its L_2-norm objective function penalizes large residuals and aligns with the assumptions of Gaussian noise. However, it may exhibit reduced robustness when confronted with non-Gaussian noise types such as large additive partial corruptions or noises follow other kinds of distribution. There are numerous types of noise patterns that in practice often originate from distributions with heavier tails compared to the Gaussian distribution, thereby challenging the efficacy of standard L_2-norm NMF in such scenarios [6].

3.3 L_1-norm Regularized Robust NMF

The L_1-norm regularized NMF [2] is particularly designed to handle data that is partially corrupted, for instance faces covered by glasses or masks, as it operates under the assumption that such corruption is significant. Its highlight compared to the standard L_2-norm NMF, is the incorporation of a noise matrix $E \in \mathbb{R}^{m \times n}$, which serves as the predicted noise that is separated from the reconstructed matrix to enable its more accurate approximation of the clean data during optimization. It is important to note that E is assumed to be modelling non-Gaussian noise, preferably sparse additive partial corruptions. To formulate this, given the contaminated data matrix as $X \in \mathbb{R}^{m \times n}$, and the clean data matrix $\hat{X} \in \mathbb{R}^{m \times n}$ being approximated by UV, where $U \in \mathbb{R}^{m \times k}$ and $V \in \mathbb{R}^{k \times n}$ as in the standard L_2-norm NMF, with the noise matrix E, the following depicts their relationship:

$$X \approx UV + E \tag{5}$$

3.3.1 Cost Function

Since the noise (i.e., partial corruptions) is typically assumed to be sparse in practice, with the above formula in mind, the objective function $O_{\text{L}_1\text{–regularized}}$ for L_1-norm regularized robust NMF is defined as follows:

$$O_{\text{L}_1\text{–regularized}} = \|X - UV - E\|_F^2 + \lambda \sum_j [\left\|E_{\cdot j}\right\|_0]^2 \tag{6}$$

Where the first term measures the reconstruction error while implicitly approximating the clean data matrix by UV. The second term is to regularize the sparseness of the noise matrix E, while the λ term is a hyper-parameter that defines the strength of this regularization, effectively allowing trade-off between precision and recall of noise prediction, while accurate noise prediction could lead to lower reconstruction error specified by the first term.

However, despite the L_0-norm is naturally associated with the concept of sparsity, it is commonly known for its difficulty to optimize due to its non-differentiability. Therefore L_1-norm is used as a replacement to approximate it, which is a commonly adopted strategy [2]. With the L_1-norm integrated into the objective function $O_{\text{L}_1\text{–regularized}}$, we arrive at:

$$\begin{aligned} O_{\text{L}_1\text{–regularized}} &= \|X - UV - E\|_F^2 + \lambda \sum_j [\left\|E_{\cdot j}\right\|_1]^2 \\ &= \left\| X - [U, I, -I] \begin{pmatrix} V \\ \begin{pmatrix} E^p \\ E^n \end{pmatrix} \end{pmatrix} \right\|_F^2 \\ &+ \lambda \sum_j \left[\left\| E^p_{\cdot j} \right\|_1 + \left\| E^n_{\cdot j} \right\|_1 \right]^2 \end{aligned} \tag{7}$$

Where $\|\cdot\|_F$ denotes the Frobenius norm, $E = E^p - E^n$, $E^p = \frac{|E|+E}{2}$, $E^n = \frac{|E|-E}{2}$ and $E^p, E^n \geq 0$.

The decomposition of E into two non-negative sub-matrices is motivated by the fact that E itself could naturally be either negative or non-negative. While it is necessary to maintain non-negativity of matrices in NMF optimization, the decomposition facilitates it and thereby making the updates convenient [2]. Bringing everything together, the aim is to minimize the above objective function in regards to the constraints of $U \geq 0$, $V \geq 0$, $E^p \geq 0$, $E^n \geq 0$ and $X - E \geq 0$, the last of which ensures that the clean data remains non-negative.

3.3.2 Optimization

Similar to the optimization for the standard L_2-norm NMF, it is impractical to find the global minima which involves optimizing U, V, E^p, E^n together. Therefore instead each is updated respectively and

iteratively, as in the standard L_2-norm NMF. The multiplicative update rules for each are defined as follows:

Update U
Given V, E^p, and E^n, the objective for U is defined as:

$$\begin{aligned} U &= \arg\min_{U\geq 0} \left\| X - [U, I, -I] \begin{pmatrix} V \\ \begin{pmatrix} E^p \\ E^n \end{pmatrix} \end{pmatrix} \right\|_F^2 \\ &\quad + \lambda \sum_j \left[\left\| E^p_{.j} \right\|_1 + \left\| E^n_{.j} \right\|_1 \right]^2 \\ &= \arg\min_{U\geq 0} \| [X - E] - UV \|_F^2 \end{aligned} \tag{8}$$

Which leads to the update rule for U that minimizes the objective function as:

$$U_{ij} = \frac{U_{ij}(\hat{X}V^T)_{ij}}{(UVV^T)_{ij}} \tag{9}$$

Where $\hat{X} = X - E$, that refers to the approximated clean data using E, while E is computed by the given E^p and E^n, and the constraint of $X - E \geq 0$ is satisfied.

Update V**,** E^p**, and** E^n
On the other hand, we minimizes the objective function with respect to V, E^p, and E^n, given U. We start by defining $\tilde{V} = \begin{pmatrix} V \\ \begin{pmatrix} E^p \\ E^n \end{pmatrix} \end{pmatrix}$ for simplicity of expression. We then have:

$$\tilde{V}_{ij} = \max\left(0, \tilde{V}_{ij} - \frac{\tilde{V}_{ij}(\tilde{U}^T\tilde{U}\tilde{V})_{ij}}{(S\tilde{V})_{ij}} + \frac{\tilde{V}_{ij}(\tilde{U}^T\tilde{X})_{ij}}{(S\tilde{V})_{ij}}\right) \tag{10}$$

Where $\tilde{X} = \begin{pmatrix} X \\ 0_{1\times n} \end{pmatrix}$, $\tilde{U} = \begin{pmatrix} U, I, -I \\ 0_{1\times k} \sqrt{\lambda} e_{1\times m} \sqrt{\lambda} e_{1\times m} \end{pmatrix}$, in which I denotes the identity matrix, and $S_{ij} = \left|(\tilde{U}^T\tilde{U})_{ij}\right|$.

3.3.3 Characteristics

The benefits of the incorporated noise matrix E by the L_1-norm regularized NMF are obvious. Not only it can be learned alongside with other factors, but it also allows the algorithm to directly account for and isolate noise without needing prior knowledge of it. This helps in mitigating the noise from affecting the reconstruction. Additionally, the noise matrix E can be integrated with other NMF algorithms that benefit from knowing the location of the noise, such as WNMF [2, 7]. However, as aforementioned the noise is often assumed to be partial corruptions, it therefore can be less robust to noises that follow other kinds of distribution. On the other hand, there is no explicit guarantee that the noise can be learned accurately under varying circumstances, therefore its robustness to different kinds of noise depend. While its hyper-parameter λ for regularizing the sparsity of the noise matrix E could be non-trivial to tune, which can be crucial for the robustness performance of the algorithm. Additionally, having the mechanism of attempting to predict noise when no noise is present could be a burden for the algorithm trying to focus on the matrix factorization.

3.4 $L_{2,1}$-norm based NMF

The $L_{2,1}$-norm NMF integrates a non-squaring property [3], unlike the standard L_2-norm NMF, which makes it less sensitive to outliers and specific types of noise.

3.4.1 Cost Function

Let $X = (x_1, \cdots, x_n)$ and $H = (h_1, \cdots, h_n)$, the cost function of the $L_{2,1}$-norm NMF is defined as follows:

$$\|X - WH\|_{2,1} = \sum_{i=1}^{n} \sqrt{\sum_{j=1}^{p} (X - WH)_{ji}^2} = \sum_{i=1}^{n} \|x_i - Wh_i\| \tag{11}$$

Which differs from the standard L_2-norm NMF by having the square root term for the inner summation, essentially prompting dominance reduction of outliers. The goal is to minimize this objective function in regards to W and H, with respect to constraints of $W \geq 0$ and $H \geq 0$.

3.4.2 Optimization

While the difficulty of optimizing both W and H together also exist here, similar to the other two NMF algorithms, the optimization is performed via a set of dedicated convergence-proven Multiplicative Update Rules (MURs), that are formulated as follows:

$$W_{jk} \leftarrow W_{jk} \frac{(XDH^T)_{jk}}{(WHDH^T)_{jk}} \tag{12}$$

$$H_{ki} \leftarrow H_{ki} \frac{(W^T XD)_{ki}}{(W^T WHD)_{ki}} \tag{13}$$

Where D is a diagonal matrix, and its diagonal entries are specified by:

$$D_{ii} = \frac{1}{\sqrt{\sum_{j=1}^{p} (X - WH)_{ji}^2}} = \frac{1}{\|x_i - Wh_i\|} \tag{14}$$

Despite the additional computation required for the term D on top of the ones for the standard L_2-norm NMF, it is shown [3] that $L_{2,1}$-norm NMF typically converges with less number of iterations relatively. Furthermore, following the proposed MURs has been proven to decrease the objective function monotonically [3] during the optimization.

3.4.3 Characteristics

The non-squaring attribute in the $L_{2,1}$-norm NMF theoretically enhances its robustness against outliers and specific types of additive noise. This is because the penalty on the residual difference between the noisy and reconstructed data has a lesser impact on the overall loss magnitude. Moreover, the $L_{2,1}$-norm NMF can be seen as inherently aligning with the Laplacian noise assumptions from the probability point of view, against which it may exhibit robustness. However, there is no theoretical assurance that it will perform well in cases of severe contamination, where its relative robustness might be negligible.

3.5 Noise description

This section describes the three noise types that we use for simulation in our experiments.

3.5.1 Gaussian noise

Gaussian noise has been ubiquitous in practice for various domains. It is a type of statistical noise, where each pixel in an image, or a data entry in general, is subjected to a random variation conforming a Gaussian distribution. The Gaussian distribution is defined by a mean and a standard deviation. The mean is commonly assumed to be zero that prompts a symmetric distribution around zero for both positive and negative values. The formula of Gaussian noise with mean μ and standard deviation σ is given by its Probability Density Function (PDF):

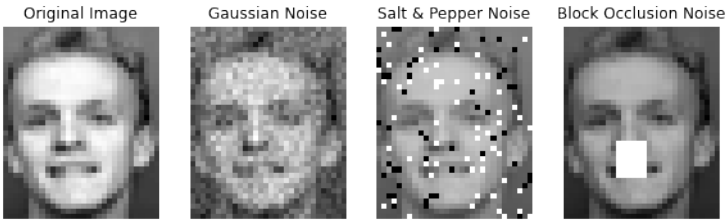


Figure 1: Demonstration of noise types: From left to right are Gaussian noise with $\sigma = 0.05$, Salt & Pepper noise with $p = 0.1$ and $salt\_p = 0.5$, and Block Occlusion noise with block width and block height both to be 20% of the original image size, respectively. Image randomly chosen from the ORL dataset.

$$f(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{15}$$

Where in our experiments which focus on face image reconstruction, at each time of experiment, a matrix of Gaussian noise with the same shape as the full data matrix is constructed by random sampling from the Gaussian distribution with a pre-defined mean and standard deviation, where each data entry (i.e., pixel) is perturbed by a distinct random scalar that is described by the above PDF formula. Figure 1 gives an example of what an image contaminated by Gaussian noise with a specific mean and standard deviation could look like. Additionally, to ensure the data after the simulation of Gaussian noise remains in valid pixel range, we utilize an implementation trick to replace any values beyond the boundary with the nearest extreme value (e.g., negative values are replaced with 0).

3.5.2 Salt & Pepper noise

The Salt & Pepper noise is also a common type in practice, for instance, bit errors during data transmission, sudden disruptions in the image signal, or faulty hardware memory locations. This type of noise in image processing manifests as sparsely occurring white and black pixels, giving the appearance of "salt and pepper", in which salt refers to white pixels and pepper refers to black. Figure 1 presents an example of the Salt & Pepper noise.

In the numerical representation of the data matrix, the Salt & Pepper noise involves setting the salt pixel to be 1 and pepper pixel to be 0, assuming pixel intensity values have been normalized to the range of [0, 1]. We typically have two adjustable factors for this noise, namely the probability p of a pixel chosen to be contaminated and the probability $salt\_p$ that defines whether a contaminated pixel should be salt or pepper. Unlike Gaussian noise, Salt & Pepper noise is an impulse-type noise that does not conform to a continuous probability distribution. It is employed to rigorously evaluate the robustness of NMF algorithms, particularly when considered alongside Gaussian noise.

3.5.3 Block Occlusion noise

Block Occlusion noise simulates the scenarios where contiguous pixels or blocks are substituted with non-contextual or predetermined values in image processing, akin to phenomena like object obstructions or faces wearing glasses. Although there is a wide range of flexibility in configuring the attributes of Block Occlusion noise, in our experiments, we opt to set the pixel intensity within a randomly selected block region to 1 in the range of [0-1], effectively making it completely white. The width and the height of the block as the adjustable factors are defined as fractions relative to the proportions of the original image dimensions. An example of Block Occlusion noise is illustrated in Figure 1. This form of noise, classified under large partial corruptions, offers an alternative perspective for evaluating the resilience of each NMF algorithm, aiding in a more thorough analysis of their robustness.

4 Experiments

4.1 Dataset

In this study, we perform our experiments based on the two widely used face data sets, ORL and Extended YaleB.

4.1.1 ORL dataset

The ORL dataset [4] is a widely used dataset of 400 images of human faces that belong to 40 distinct subjects, each represented by 10 images. These images were captured at varied instances, incorporating different lighting conditions, facial expressions and external details, while the photographic setting is consistent. Each image comes as 92x112 pixels, as mentioned in the pre-processing part, we then further reduce the image size by 3 times to 30x37 pixels.

4.1.2 Extended YaleB dataset

The extended version of another famous dataset of human faces named YaleB dataset [5] is used in our experiments. This dataset is larger and contains 2414 images of 38 subjects, taken under 9 poses and 64 illumination conditions. The images come with a shape of 168×192 pixels, and are also further reduced, by 4 times to 42x48 pixels during our experiments.

4.2 Metrics

We evaluate the performance of the NMF models based on three metrics, namely the Relative Reconstruction Error, Averaged Accuracy and normalized Mutual Information. The latter two metrics are calculated based on class predictions generated via K-means clustering on the transformed data matrix, with majority voting for each cluster. These metrics are commonly used in the field of NMF studies, and are defined as follows:

- Relative Reconstruction Error (RRE):

$$\text{RRE} = \frac{\|\hat{V} - W \times H\|_{\text{F}}}{\|\hat{V}\|_{\text{F}}} \tag{16}$$

Where $\hat{V}$ defines the non-contaminated original data matrix, $W \times H$ defines the reconstructed data matrix learned from the contaminated dataset. RRE measures the residual between the original and reconstruction while normalized on the original clean data that mitigates the sensitivity to the scale of the data.

- Averaged Accuracy (ACC):

$$\text{ACC}(Y, Y_{\text{pred}}) = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}(Y_{\text{pred}}(i) = Y(i)) \tag{17}$$

To obtain the label predictions, we perform K-means clustering on the transformed data matrix H, derived from the decomposition $V \approx W \times H$, and then employ majority voting to assign a representative label to each cluster. The number of clusters is set to the number of subjects in the respective dataset. We then measure the accuracy performance of the clustering, that gives us roughly the degree of corresponding informational congruence between the original data and the transformed data.

- normalized Mutual Information (NMI):

$$\text{NMI}(Y, Y_{\text{pred}}) = \frac{2 \times \text{I}(Y, Y_{\text{pred}})}{\text{H}(Y) + \text{H}(Y_{\text{pred}})} \tag{18}$$

Where $I(\cdot, \cdot)$ defines the mutual information and $H(\cdot)$ refers to the entropy term. The predictions are obtained via the same procedure as the averaged accuracy.

4.3 Choice of the number of components

While the performance of each NMF algorithm varies on its inherent characteristics, the number of components (often denoted as k) as a critical hyper-parameter could significantly influence the performance result, and is often non-trivial to optimize. For the scope of this study, we evaluate a range of k on the ORL dataset [4], specifically from 10 to 80 with a step size of 10, considering both non-contaminated and Salt & Pepper noise-contaminated scenarios, to observe the performance variations for this particular task.

Figure 2 shows the Relative Reconstruction Error (RRE) of the NMF algorithms on respective settings. A larger number of components k generally results in a more accurate fit to the training data due to the increased parameter space. It therefore presents a consistently declining trend of the RRE for all the NMF algorithms if the data is not contaminated. While the context of these NMF experiments does not involve an unseen distribution, it is worth noting that potential overfitting could still exist, especially with larger k, affording the algorithms greater capacity to encapsulate specific idiosyncrasies from the given data. Additionally, it is noted that there is a shrinking gesture of the RRE decrement as k goes larger, that is more explicit for L_1-norm regularized NMF and $L_{2,1}$-norm NMF. While on the other hand, in a contaminated data setting, a larger k may lead to capturing the noise characteristics, particularly in the absence of effective regularization methods, therefore misleading the overall learning, as evidenced by the upward trajectory of RRE in the right panel of Figure 2. As a side note from that figure, the L_1-norm regularized NMF consistently gives a lower RRE on this particular noise setting that demonstrates relative robustness over the other two NMF algorithms across different k.

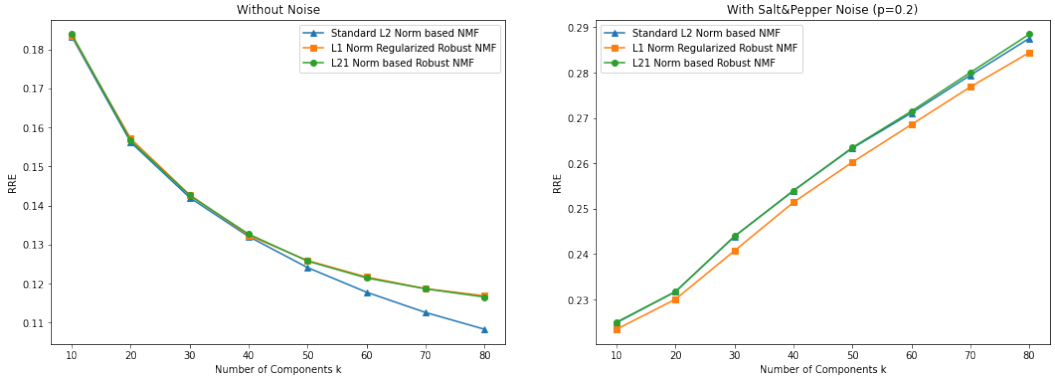


Figure 2: RRE Performance of NMF Algorithms over a Range of k in both data contaminated and non-contaminated settings.

4.4 Algorithm Settings

As a result of our tests on the number of components k, we select $k = 60$ for the subsequent experiments for a reasonably balanced performance with relative computational efficiency. We also fix our maximum number of iterations for all NMF algorithms across all experiments on both datasets to be 500. It is worth noting that there is another hyper-parameter λ for the L_1-norm regularized NMF, which is a positive constant coefficient that defines the strength of the L_1-norm regularization on the learned noise matrix, that trades off the sparsity of the predicted noise. It can be treated as how confident the model should be on predicting a pixel as contaminated while benefiting from less reconstruction error if correctly predicted on the other hand. Given its complexity, it is often non-trivial to optimally select λ in practice. For all of our experiments, we set $\lambda = 0.04$ according to the original paper [2], in which the selection is based on a relevant study conducted that presents a reasonable balance between the precision and recall on the noise prediction. In terms of the weight initialization, for all NMF algorithms, we employ non-negative random weight sampling from the standard normal distribution, scaled by a factor of $\sqrt{\frac{\text{mean}(\hat{V})}{k}}$, where $\hat{V}$ refers to the original clean data and k refers to the number of components.

4.5 Noise Settings

We test varying noise strengths for each of the noise types applied on the NMF algorithms, on both datasets for our experiments. The purpose is to observe and compare how the NMF algorithms react to different degrees of severity of noises in practice. Figure 3 gives an example for each contamination case on a randomly sampled image from the ORL dataset. For Gaussian noise, we fix the mean μ to 0 and test on the σ range of 0.03 to 0.15 with a step size of 0.03. While the Salt & Pepper noise has two arguments (i.e., the probability p of a pixel being contaminated and the probability *salt_p* of a pixel being absolutely white (salt) or conversely *1 - salt_p* of being absolutely black (pepper) if it is chosen to be contaminated), we fix *salt_p* to 0.5, and traverse the p from 0.05 to 0.25 with a step size of 0.05. Lastly for Block Occlusion noise, we define the width and height of the block to be proportional to the original image size, while making the proportion for the width and height of the block according to the original image the same for simplicity, that we will call block size. We use block size of 10% (0.1) of the original image size to 30% (0.3) for the experiments, with a step size of 5% (0.05).

(a) (b) (c)

Figure 3: Demo of the used noise strengths for each corresponding noise type in the experiments. From left to right are Gaussian noise (a), Salt & Pepper noise (b) and Block Occlusion noise (c) respectively, with the noise strength increasing from left to right, top to bottom.

Experiment	RRE ↓	ACC ↑	NMI ↑
k = 60 | No. of iterations = 500			
Standard L_2-NMF	**0.1127** (0.0005)	0.7283 (0.0123)	0.8469 (0.0111)
L_1-norm regularized NMF	0.1132 (0.0004)	0.7350 (0.0378)	0.8522 (0.0178)
$L_{2,1}$-NMF	0.1134 (0.0005)	**0.7472** (0.0153)	**0.8574** (0.0101)
k = 200 | No. of iterations = 500			
Standard L_2-NMF	**0.0698** (0.0002)	0.6817 (0.0273)	**0.8236** (0.0174)
L_1-norm regularized NMF	0.0706 (0.0002)	**0.6827** (0.0299)	0.8209 (0.0158)
$L_{2,1}$-NMF	0.0710 (0.0002)	0.6600 (0.0176)	0.8046 (0.0089)
k = 200 | No. of iterations = 3000			
Standard L_2-NMF	**0.0562** (0.0004)	**0.6161** (0.0361)	**0.7733** (0.0217)
L_1-norm regularized NMF	0.0569 (0.0003)	0.6050 (0.0189)	0.7610 (0.0138)
$L_{2,1}$-NMF	0.0571 (0.0004)	0.5972 (0.0165)	0.7537 (0.0192)

Table 1: Performance - {Mean (Standard Deviation)} of the NMF algorithms on the ORL dataset without contamination.

4.6 Results

We firstly test the NMF algorithms on the ORL dataset without contamination, aiming to compare both quantitatively and qualitatively the ability of each NMF algorithm in approximating the target

data matrix. We then run the NMF algorithms on both datasets with the aforementioned settings to assess and evaluate their respective robustness against various noises in the data. In order to ensure the reliability of the outcome of our experiments, we take the final result for each experimental case as the mean across 5 instances, with each instance random sampling 90% of the data from the original dataset. We also report the standard deviations to give a sense of the stability of the NMF algorithms under the particular settings.

4.6.1 Performance on the ORL dataset without contamination

To better analyze particularly this task, we additionally assess the NMF algorithms with larger number of components k and number of iterations, specifically we add two extra experiment cases: 1. k = 200 with number of iterations = 500, and 2. k = 200 with number of iterations = 3000, on top of the default setting: k = 60 and number of iterations = 500, that gives an estimation of the depth of the approximation ability of each NMF algorithm.

Table1 shows the results of each NMF algorithm trained under each case. It is not surprising that the standard L_2-NMF has better RRE under all the cases, since its L_2-norm that squares the residuals typically could give a closer fit on the target data by penalizing significant errors, while it does not have the burden of carrying additional learnable parameters as the L_1-norm regularized NMF, which are not meaningful when data contamination is not present, that could potentially negatively impact the dynamics of convergence in this specific context. Nevertheless, while figure 4 shows the reconstruction outputs on a randomly sampled ORL image by each NMF algorithm under each test case respectively, there is no significant performance gap visually speaking, and that all three NMF algorithms demonstrate increasing fidelity in reconstructing images that closely approximate the target image as the k and number of iterations increase.

(a) (b) (c)

Figure 4: Reconstruction of a random ORL image by Standard L_2-norm NMF (a), L_1-norm regularized NMF (b) and $L_{2,1}$-norm NMF (c) from left to right, respectively. On the top left is the actual image, the top right is the result trained under (k = 60, No. of iteration = 500), and (k = 200, No. of iteration = 500), (k = 200, No. of iteration = 3000) for bottom left and bottom right, respectively.

4.6.2 Performance on both datasets with contamination

We perform a more extensive set of experiments based on the aforementioned settings on both datasets that focus on the robustness of each NMF algorithms against various noise scenarios.

Figure 5 shows the performance of each NMF algorithms on the ORL dataset when the images are contaminated with Gaussian noises. We can see the RRE of all the algorithms consistently increase as the noise strength goes up, while the rate of increase becomes noticeably larger at higher Gaussian noise strengths, this may suggest these NMF algorithms could be sensitive to significant contamination. Although there seems to be no explicit correlation from the accuracy and NMI visually, we can still observe the decreasing trends of all NMF algorithms as the noise strength increases.

On the other hand, the standard deviations in the context of Gaussian noise seem obvious in the visual perspective, however we attribute this to the scale of the y-axis, while the values are actually quite similar compared to the settings with other noises in Table 3 that are not visually obvious. The standard deviations of these NMF algorithms under all the experimented noises in Table 3 are actually acceptable and are considered small in terms of the Coefficient of Variation where we compare with

the mean. While each experimental instance not only differs in the set of base images, but also the noise distribution, that contribute to the differences in the distribution of training data. Therefore overall suggesting these NMF algorithms tend to be stable even under noises.

The standard L_2-norm NMF has demonstrated generally better performance than the others throughout the experiments with Gaussian noise, especially for RRE, as demonstrated in table 2. This aligns with its theoretical assumption that noise follows the Gaussian distribution [6, 3], and so the standard L_2-norm NMF is relatively robust to Gaussian noise when considering the reconstruction. Conversely, the L_1-norm regularized NMF comparatively demonstrates least effective performance, that could be attributed to its inherent focus on large partial corruptions, coupled with its assumption that the additive noise is non-Gaussian in practice [2]. Specifically, the introduced error matrix E which as the essence of the L_1-norm regularized NMF on top of the standard NMF, while it is designed to predict the position of the noises, may struggle to effectively capture the noise patterns in scenarios where each data entry is subject to some level of Gaussian noise perturbation. Additionally, the L_1 regularization term that encourages the sparsity of E, could further prevents the algorithm from appropriately modelling the Gaussian noise.

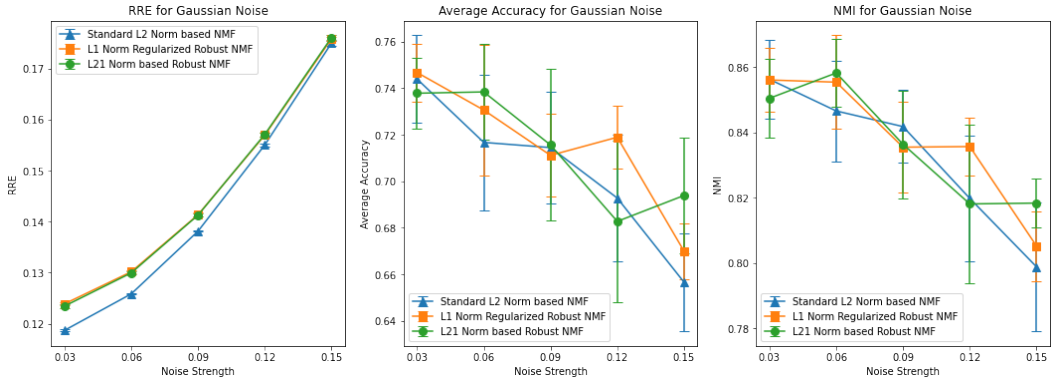


Figure 5: Performance of the NMF Algorithms on the ORL datasets with varying degrees of Gaussian Noise.

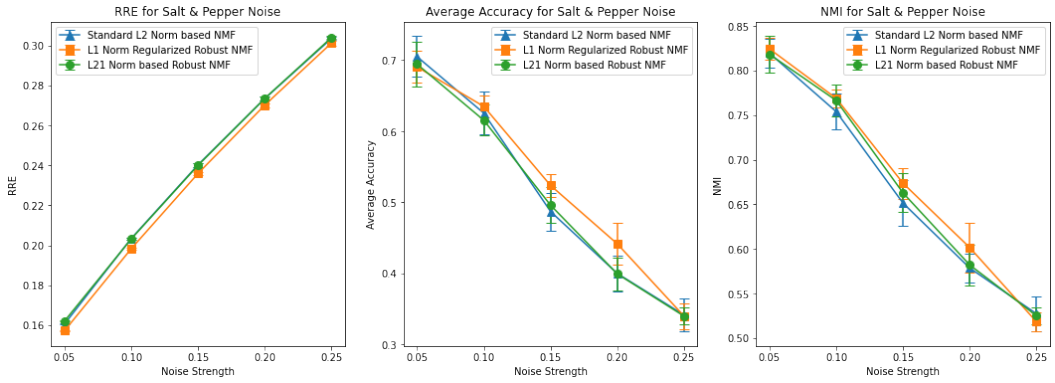


Figure 6: Performance of the NMF Algorithms on the ORL datasets with varying degrees of Salt & Pepper Noise.

However, in the context of Salt & Pepper noise or Block Occlusion noise, which both are considered a sub-class of partial corruptions, the L_1-norm regularized NMF in many cases outperforms the others across both datasets, especially consistently in terms of RRE, as evident by the results in Table 2. The explicit RRE gaps in Figure 6 and Figure 7 further emphasizes the reconstruction advantages of the L_1-norm regularized NMF compared to the others, this demonstrates the relative effectiveness of the L_1-regularized error matrix E. Additionally, the visualization of some examples of the Block Occlusion noise predicted in Figure 8, shows E can quite precisely locate the position of the noise,

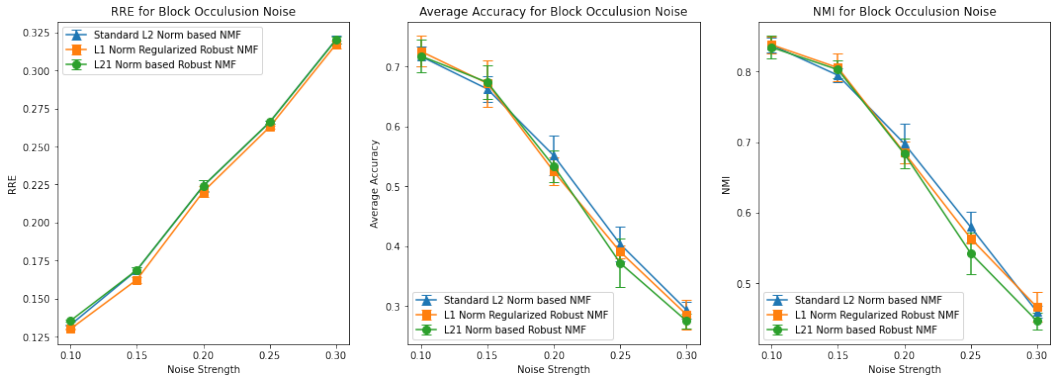
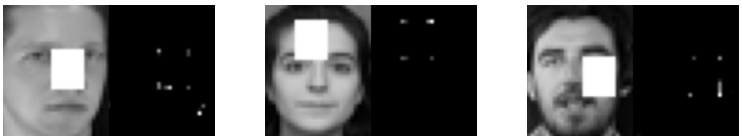


Figure 7: Performance of the NMF Algorithms on the ORL datasets with varying degrees of Block Occlusion Noise.

demonstrated by the depicted white contours of the blocks, that highly match the actual position of the block noise in the original image. Although it is not able to identify the whole block as noise, it is already quite beneficial as demonstrated by the performance results as compared to the rest NMF algorithms.

The $L_{2,1}$-norm NMF as in Table 2, although is less sensitive to some particular noises with its non-squaring property compared to the standard L_2-norm NMF, often shows sub-optimal performance compared to the others, since the experiments are comparative under specific settings, while some noises are more nature to the others (e.g., Gaussian noise for standard L_2-norm NMF), its advantages are therefore less obvious. However, there are some noises such as Laplacian noise [3] where the $L_{2,1}$-norm NMF is more preferred, which may be left to future work.

Additionally, we further qualitatively assess the reconstruction quality of these NMF algorithms under the Salt & Pepper noise, as shown in Figure 9. We can see that under small amount of noise (i.e., $p = 0.05$), all algorithms are able to largely revert the original images. However, the standard L_2-norm NMF and the $L_{2,1}$-norm NMF tend to have the noise incorporated in the reconstruction, while the L_1-norm regularized NMF presents noise-less reconstructed image, which convincingly demonstrates the relative robustness of the L_1-norm regularized NMF over the other two algorithms under small noises of partial corruptions. Nonetheless, all three algorithms fail to reconstruct identifiable image when the noise strength is high (i.e., $p = 0.25$), that suggests the infeasibility of these NMF algorithms when the data is significantly contaminated, that is also evident by the large performance drops in Figure5, 6 and 7.

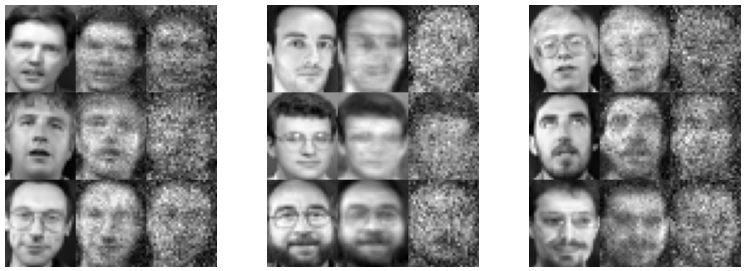


(a) (b) (c)

Figure 8: Three examples of the noise position prediction by the error matrix E from the L_1-norm regularized NMF in the context of Block Occlusion noise with size = 0.3. On the left is the contaminated noise, on the right is the predicted noise depicted by white pixels.

5 Conclusion

In this paper, we have examined the performance of three specific NMF algorithms, namely standard L_2-norm NMF, L_1-norm regularized NMF and $L_{2,1}$-norm NMF. The evaluations were carried out under predefined experimental conditions, across the ORL [4] and Extended YaleB [5] datasets under both non-contaminated and contaminated scenarios subject to three types of noises. The study



(a) (b) (c)

Figure 9: Demo of image reconstruction by Standard L_2-norm NMF, L_1-norm regularized NMF and $L_{2,1}$-norm NMF from left to right respectively, under the Salt & Pepper noise, where the first column are original images, the second column are $p = 0.05$ and third column are $p = 0.25$.

Experiment	ORL			Extended YaleB		
Gaussian Noise | $\sigma = 0.03$	RRE ↓	ACC ↑	NMI ↑	RRE ↓	ACC ↑	NMI ↑
Standard L_2-NMF	**0.1187**	0.7439	**0.8563**	**0.1763**	**0.2544**	**0.3321**
L_1-norm regularized NMF	0.1239	**0.7467**	0.8561	0.1774	0.2512	0.3288
$L_{2,1}$-NMF	0.1235	0.7378	0.8505	0.1812	0.2409	0.3172
Gaussian Noise | $\sigma = 0.09$	RRE ↓	ACC ↑	NMI ↑	RRE ↓	ACC ↑	NMI ↑
Standard L_2-NMF	**0.1381**	0.7144	**0.8418**	**0.1851**	0.2474	**0.3291**
L_1-norm regularized NMF	0.1414	0.7111	0.8355	0.1859	0.2445	0.3204
$L_{2,1}$-NMF	0.1413	**0.7156**	0.8364	0.1877	**0.2517**	0.3244
Gaussian Noise | $\sigma = 0.15$	RRE ↓	ACC ↑	NMI ↑	RRE ↓	ACC ↑	NMI ↑
Standard L_2-NMF	**0.1750**	0.6567	0.7989	**0.2056**	0.2320	**0.3127**
L_1-norm regularized NMF	0.1758	0.6700	0.8052	0.2062	0.2362	0.3108
$L_{2,1}$-NMF	0.1761	**0.6939**	**0.8184**	0.2069	**0.2370**	0.3122
Salt & Pepper Noise | p = 0.05	RRE ↓	ACC ↑	NMI ↑	RRE ↓	ACC ↑	NMI ↑
Standard L_2-NMF	0.1606	**0.7050**	0.8194	0.1939	**0.2366**	0.3156
L_1-norm regularized NMF	**0.1572**	0.6906	**0.8242**	**0.1923**	0.2339	0.3165
$L_{2,1}$-NMF	0.1621	0.6944	0.8181	0.1968	0.2321	**0.3214**
Salt & Pepper Noise | p = 0.15	RRE ↓	ACC ↑	NMI ↑	RRE ↓	ACC ↑	NMI ↑
Standard L_2-NMF	0.2403	0.4872	0.6514	0.2613	**0.2136**	0.2822
L_1-norm regularized NMF	**0.2359**	**0.5239**	**0.6736**	**0.2586**	0.2099	**0.2881**
$L_{2,1}$-NMF	0.2400	0.4961	0.6628	0.2622	0.2111	0.2803
Salt & Pepper Noise | p = 0.25	RRE ↓	ACC ↑	NMI ↑	RRE ↓	ACC ↑	NMI ↑
Standard L_2-NMF	0.3036	**0.3417**	**0.5273**	0.3477	0.1722	0.2250
L_1-norm regularized NMF	**0.3013**	0.3400	0.5183	**0.3447**	0.1711	**0.2322**
$L_{2,1}$-NMF	0.3040	0.3400	0.5250	0.3473	**0.1743**	0.2228
Block Occlusion Noise | size = 0.1	RRE ↓	ACC ↑	NMI ↑	RRE ↓	ACC ↑	NMI ↑
Standard L_2-NMF	0.1327	0.7167	0.8376	0.1966	0.2260	0.3033
L_1-norm regularized NMF	**0.1301**	**0.7250**	**0.8381**	**0.1897**	**0.2431**	**0.3085**
$L_{2,1}$-NMF	0.1355	0.7172	0.8344	0.1980	0.2367	0.3041
Block Occlusion Noise | size = 0.2	RRE ↓	ACC ↑	NMI ↑	RRE ↓	ACC ↑	NMI ↑
Standard L_2-NMF	0.2241	**0.5511**	**0.6979**	0.3841	0.1476	0.1900
L_1-norm regularized NMF	**0.2203**	0.5244	0.6849	**0.3800**	0.1525	0.2051
$L_{2,1}$-NMF	0.2241	0.5333	0.6833	0.3830	**0.1550**	**0.2063**
Block Occlusion Noise | size = 0.3	RRE ↓	ACC ↑	NMI ↑	RRE ↓	ACC ↑	NMI ↑
Standard L_2-NMF	0.3203	**0.2933**	0.4592	0.5807	0.1043	0.1117
L_1-norm regularized NMF	**0.3172**	0.2856	**0.4657**	**0.5782**	0.1010	0.1096
$L_{2,1}$-NMF	0.3201	0.2750	0.4460	0.5797	**0.1049**	**0.1162**

Table 2: **Mean** value results across 5 instances of each experiment on the ORL and Extended YaleB datasets.

Experiment	ORL			Extended YaleB		
Gaussian Noise | σ = 0.03	RRE	ACC	NMI	RRE	ACC	NMI
Standard L_2-NMF	**0.0001**	0.0187	0.0120	0.0006	**0.0028**	0.0086
L_1-norm regularized NMF	0.0003	**0.0125**	**0.0098**	**0.0003**	0.0155	0.0149
$L_{2,1}$-NMF	0.0003	0.0153	0.0122	0.0005	0.0096	**0.0071**
Gaussian Noise | σ = 0.09	RRE	ACC	NMI	RRE	ACC	NMI
Standard L_2-NMF	**0.0001**	0.0240	**0.0111**	0.0006	0.0114	**0.0078**
L_1-norm regularized NMF	0.0004	**0.0177**	0.0140	**0.0004**	0.0129	0.0162
$L_{2,1}$-NMF	0.0003	0.0325	0.0165	**0.0004**	**0.0073**	0.0146
Gaussian Noise | σ = 0.15	RRE	ACC	NMI	RRE	ACC	NMI
Standard L_2-NMF	0.0005	0.0210	0.0197	0.0006	0.0113	0.0079
L_1-norm regularized NMF	**0.0004**	**0.0120**	0.0107	**0.0004**	0.0036	0.0087
$L_{2,1}$-NMF	**0.0004**	0.0247	**0.0076**	0.0005	**0.0028**	**0.0066**
Salt & Pepper Noise | p = 0.05	RRE	ACC	NMI	RRE	ACC	NMI
Standard L_2-NMF	**0.0003**	0.0286	0.0157	0.0005	0.0081	**0.0114**
L_1-norm regularized NMF	**0.0003**	**0.0221**	**0.0124**	**0.0003**	**0.0066**	0.0136
$L_{2,1}$-NMF	**0.0003**	0.0320	0.0203	**0.0003**	0.0073	0.0126
Salt & Pepper Noise | p = 0.15	RRE	ACC	NMI	RRE	ACC	NMI
Standard L_2-NMF	0.0010	0.0266	0.0253	**0.0008**	0.0141	0.0112
L_1-norm regularized NMF	**0.0007**	**0.0165**	**0.0171**	0.0012	**0.0045**	0.0143
$L_{2,1}$-NMF	0.0010	0.0250	0.0217	0.0009	0.0064	**0.0082**
Salt & Pepper Noise | p = 0.25	RRE	ACC	NMI	RRE	ACC	NMI
Standard L_2-NMF	0.0008	0.0229	0.0192	0.0009	0.0084	0.0108
L_1-norm regularized NMF	0.0013	0.0181	0.0106	0.0014	**0.0043**	**0.0072**
$L_{2,1}$-NMF	**0.0006**	**0.0117**	**0.0096**	**0.0008**	0.0123	0.0110
Block Occlusion Noise | size = 0.1	RRE	ACC	NMI	RRE	ACC	NMI
Standard L_2-NMF	0.0005	**0.0173**	**0.0110**	0.0007	**0.0064**	0.0169
L_1-norm regularized NMF	**0.0004**	0.0256	0.0117	**0.0005**	0.0119	0.0115
$L_{2,1}$-NMF	0.0006	0.0272	0.0160	0.0008	0.0080	**0.0029**
Block Occlusion Noise | size = 0.2	RRE	ACC	NMI	RRE	ACC	NMI
Standard L_2-NMF	0.0037	0.0326	0.0277	0.0013	0.0072	**0.0101**
L_1-norm regularized NMF	**0.0034**	**0.0229**	**0.0153**	0.0013	**0.0062**	0.0152
$L_{2,1}$-NMF	0.0035	0.0266	0.0210	**0.0012**	0.0082	0.0153
Block Occlusion Noise | size = 0.3	RRE	ACC	NMI	RRE	ACC	NMI
Standard L_2-NMF	**0.0023**	0.0143	**0.0084**	**0.0017**	**0.0030**	**0.0050**
L_1-norm regularized NMF	0.0024	0.0249	0.0209	0.0018	**0.0030**	0.0079
$L_{2,1}$-NMF	0.0024	**0.0123**	0.0120	0.0024	0.0041	0.0073

Table 3: **Standard Deviations** across 5 instances of each experiment on ORL and Extended YaleB datasets.

shows larger number of components k could lead to better performance of the algorithms, but may subject to potential overfitting especially when noise is present, while the performance increase rate could saturate as k increases as evident in Figure 2. Both quantitative and qualitative results show all three NMF algorithms are capable of presenting reasonably good reconstruction when the data is not contaminated. When considering noises, the standard L_2-norm NMF is more robust towards the Gaussian noise, while the L_1-norm regularized NMF is particularly robust against the Salt & Pepper noise and Block Occlusion noise that belong to the class of partial corruptions, over the other two algorithms. The $L_{2,1}$-norm NMF often presents sub-optimal performance comparatively in this particular setting but may demonstrate superiority under other noise settings. While all three algorithms are to some extent robust to minor noise, particularly the L_1-norm regularized NMF in the context of partial corruptions, they all encounter difficulties in reconstructing appropriate images when faced with substantial noise. On the other hand, the performance gaps among these NMF algorithms seem to be negligible when taking other relatively state-of-the-art NMF models into account [6].

6 Future Work

This study is only limited to the predefined experimental contexts on the nominated NMF algorithms, hence assessing other types of noise on these algorithms, or compare with other kinds of NMF algorithms could be meaningful to present a more comparative result and therefore better understand the superiority of each algorithm, which can be left to future work. On the other hand, different weight initialization for each NMF algorithm could play a significant role in its performance, while only one type of random sampling technique is used throughout our experiments for initializing the weights, one could assess the effectiveness of various types of weight initialization in different contexts for future work, such as the SVD-based initialization [8]. There is also an opportunity to systematically explore the effect of the hyper-parameter λ for the L_1-norm regularized NMF that could potentially increase its performance in some particular scenarios. Additionally, the assessment of robustness on these NMF algorithms could be placed on other datasets with different characteristics or even beyond image reconstruction in the future.

References

[1] D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in Neural Information Processing Systems*, vol. 13, MIT Press, 2000.

[2] B. Shen, B. Liu, Q. Wang, and R. Ji, "Robust nonnegative matrix factorization via l1 norm regularization by multiplicative updating rules," in *IEEE International Conference on Image Processing (ICIP)*, (Paris, France), pp. 5282–5286, 2014.

[3] D. Kong, C. Ding, and H. Huang, "Robust nonnegative matrix factorization using l21-norm," in *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, (New York, NY, USA), p. 673–682, Association for Computing Machinery, 2011.

[4] F. Samaria and A. Harter, "Parameterisation of a stochastic model for human face identification," in *IEEE Workshop on Application and Computer Vision*, pp. 138–142, 1994.

[5] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Trans. Pattern Anal. Mach. Intelligence*, vol. 23, no. 6, pp. 643–660, 2001.

[6] N. Guan, T. Liu, Y. Zhang, D. Tao, and L. S. Davis, "Truncated cauchy non-negative matrix factorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 1, pp. 246–259, 2019.

[7] S. Zhang, W. Wang, J. C. Ford, and F. Makedon, "Learning from incomplete ratings using non-negative matrix factorization," in *SDM*, 2006.

[8] C. Boutsidis and E. Gallopoulos, "Svd based initialization: A head start for nonnegative matrix factorization," *in Pattern recognition*, p. 1350–1362, 2008.