Analysis of Non-Negative Matrix Factorization

Group members: Shiwen Xu(520569045, shxu4542), Kane Wang(530098966, qwan2318)

Abstract

In this study, we aim to study the robustness of two algorithms, when the dataset is contaminated by a large magnitude of noise or corruption. We undertake this investigation by implementing and comparing two variants of NMF algorithm, $L_{2,1}$-Norm NMF and L_1-Norm Regularized Robust NMF. The rest of the paper is structured as follows. Section 2 provides a concise overview of the relevant prior research work in the field of Non-negative Matrix Factorization. In section 3, we introduce the details of two NMF algorithms from the perspective of cost functions, optimization methods, as well as two different types of noise generation methods. Section 4 elaborates the details of experiment setups, experimental outcomes, performance comparisons and analysis, followed by the key conclusion summarized in section 5.

1 Introduction

Non-negative Matrix Factorization (NMF) is a bounded constrained optimization problem aimed at decomposing a matrix (V) into two non-negative matrices, W and H, with the constraint that all three matrices contain non-negative elements. NMF has found wide-ranging applications, including face recognition [6], motion segmentation [1], object tracking [16], and more, demonstrating significant success in these domains. However, obtaining clean datasets in real-world scenarios is often challenging, as data are frequently contaminated by a large magnitude of noise or corruption. To assess the practicality of NMF algorithms in real-world contexts, our objective is to analyze and compare various existing NMF algorithms in terms of their robustness against noise. Among the popular NMF algorithms including NMF algorithms including L_1-Norm Regularised Robust NMF, Hypersurface cost based NMF, etc, we chose to implement $L_{2,1}$-Norm NMF [10] and L_1-Norm Regularized Robust NMF [15].

Both [10] and [15] have highlighted a notable limitation of the standard Non-Negative Matrix Factorization (NMF) approach, which is its vulnerability to outliers due to its reliance on the least square error function. In response to this drawback, $L_{2,1}$ NMF method [10] replaces the summation of squared errors with a summation of residuals (aka l2 norm). This strategic adjustment is made with the specific intent of mitigating the large errors due to outliers, ensuring that they do not dominate the objective function as they are not squared in this case. L_1-Norm-regularized NMF [15] introduces an innovative approach to approximating clean data by incorporating the error matrix into the equation while jointly estimating both the values and positions of the noise. Experiments are carried out with two algorithms on Orl and YaleB contaminated by Block Noise and Pepper and Salt Noise. Different λ coefficients of L_1-Norm-regularized NMF are experimented to inspect the corresponding performance differences. Generally, the experimental results show that L_1-Norm Regularised Robust NMF generally outperforms L_2-Norm NMF.

2 Related Work

Many variants of NMF have been proposed from various perspectives, including different cost functions, optimization methods, and regularization techniques. In this section, we will discuss these variants separately.

2.1 Sparsity Constraint

Several sparse constraints, including L_0, L_1, and L_2 norm-based regularization, can be incorporated into NMF to enforce the sparsity of the two factorized matrices. The objective is to ensure that only a limited number of units, out of a large proportion, effectively represent typical data vectors. As a result, most units have values close to zero, while only a few have significantly non-zero values. In addition to the fundamental regularization method, [8, 9] introduced new algorithms for NMF with sparsity constraints, allowing for the explicit enforcement of the desired degree of sparsity.

2.1.1 Robustness to Noise

To address robustness to outliers, [7] introduced Robust Nonnegative Matrix Factorization (RNMF), which aims to obtain basis vectors that are not influenced by outliers. The authors pointed out that the least squares method provided by standard NMF is not robust to outliers when producing approximations. Drawing inspiration from the robust hypersurface cost function proposed by [14], [7] minimizes the hypersurface cost function while subjecting it to non-negativity constraints on W and H, updating them alternately. Instead of seeking a substitution for the least square method, [5] employed the Trimmed Square approach, similar to robust SVD [13]. They maintained an outlier list by identifying the most discordant observations and continuously updated this list throughout the factorization process. The identified outliers from the fitting process.

2.1.2 Optimization Method

There are three main optimization methods commonly used in Nonnegative Matrix Factorization (NMF): Alternating Nonnegative Least Squares, Multiplicative Update Method, and Gradient Approaches. Alternating Nonnegative Least Squares is the optimization method used in standard NMF [12]. In each iteration step, it solves a non-negative least squares problem to update one factor and then alternates to update the other factor. However, it generally demands more computational resources and takes longer to converge. Later, Lee and Seung introduced the Multiplicative Update Method [11], claiming faster convergence. This method has been widely adopted in many NMF variants. Additionally, Gradient Approaches were proposed by [2], reformulating the bounded constraint equation into an unconstrained optimization problem.

Overall, NMF is a popular method and widely applied in many fields such as [6, 1, 16] before the emergence of deep neural networks.

3 Methods

3.1 L2,1-Norm Based NMF

3.1.1 Cost Function

$L_{2,1}$-Norm NMF [10] is similar to the standard L_2-Norm based NMF, which does not have a regularisation term or a noise learning term in the cost function formula. The only difference of $L_{2,1}$-Norm NMF's cost function is that it sums residuals(aka L2 Norm) instead of squared residuals.

$$\|X - FG\|_{2,1} = \sum_{i=1}^{n} \sqrt{\sum_{j=1}^{p} (X - FG)_{ji}^2} = \sum_{i=1}^{n} \|X_i - FG_i\| \tag{1}$$

3.1.2 Optimisation Method

The computational iteratively updating algorithm of $L_{2,1}$-Norm NMF is

$$F_{jk} \Leftarrow F_{jk} \frac{(XDG^T)_{jk}}{(FGDG^T)_{jk}}, \tag{2}$$

$$G_{ki} \Leftarrow G_{ki} \frac{(F^T XD)_{ki}}{(F^T FGD)_{ki}}, \tag{3}$$

where D is a diagonal matrix given by,

$$D_{ii} = \frac{1}{\sqrt{\sum_{j=1}^{p} (X - FG)_{ji}^2}} = \frac{1}{\|x_i - Fg_i\|}. \tag{4}$$

The optimisation iteration will stop when it reaches the maximal iteration steps or both the dictionary and the new representation hardly change anymore.

This optimisation method guarantees the global optimal solution. Through the experiments, we found that sometimes the$L_{2,1}$-Norm NMF model reaches local optimal solution, but eventually it will converge to the global optimal solution.

3.1.3 Robustness

Different from the cost function of the standard L_2-Norm NMF which squares the residual of every data element, the cost function of $L_{2,1}$-Norm NMF does not take squares of the residuals. As a result, to some extent, $L_{2,1}$-Norm NMF does not penalise the outliers as much as standard L_2-Norm NMF does. In other words, the outliers have less impact to the $L_{2,1}$-Norm NMF model. Thus, theoretically, $L_{2,1}$-Norm NMF is more robust to noise than standard L_2-Norm NMF.

However, due to that $L_{2,1}$-Norm NMF does not have a noise learning term, although it is relatively more robust than standard L_2-Norm NMF, it cannot learn the error noise. Consequently, $L_{2,1}$-Norm NMF might not be as robust as L_1-Norm Regularised Robust NMF having error term in its cost function to noise especially when the noise is easy to learn and remove.

In addition, the $L_{2,1}$-Norm NMF does not have any regularisation terms. For example, it does not have L_1-Norm regularisation term which is a sparsity constraint. As a result, we can't control the robustness of $L_{2,1}$-Norm NMF.

3.2 L1-Norm Regularized Robust NMF

We also employ another approach known as L1-norm regularized RobustNMF [15]. This method distinguishes itself from other existing variants of NMF by addressing issues related to the unknown position of significant additive noise within the image. In contrast, previous methods primarily concentrate on dealing with the explicit noise position. In the subsequent subsection, we will delve into more specific details regarding noise generation, the cost function, the optimization function, and the regularization method. Note that the equations and derivations in section 3.2 are from [15].

3.2.1 Cost Function

Rather than factorizing the two matrices, i.e., the basis matrix and the coefficient matrix, to minimize the reconstruction error as is typical in most NMF algorithms, the authors take a distinct approach by constructing an error matrix to represent the noise. The cost function generally consists of two components: data fitting and sparse regularization.

$$O_{RobustNMF} = ||X - UV - E||_F^2 + \lambda \sum_j [||E_{\cdot j}||_0]^2 \tag{5}$$

In the data fitting part, X represents the observed corrupted data, while $\hat{X}$ signifies the clean data, and they define X = $\hat{X}$ + E. The primary objective of this component is to minimize the reconstruction error. It does so by approximating the clean data matrix using the product of UV and aims to estimate both the positions and values of the noise (E matrix). This effectively aligns with the definition that X = $\hat{X}$ + E, where $\hat{X}$ is approximated by UV.

The second component introduces a regularization term into the equation. Given the objective of modeling partial corruption in the image, where the majority of entries are expected to be zero, the L_0 norm is utilized to count the number of non-zero elements in the matrix. In other words, the second component is to constraint the sparseness of of E matrix. The choice of λ, which represents the regularization hyperparameter, controls the trade-off between reconstruction error and sparsity of the noise matrix. A larger λ value results in a sparser noise matrix, potentially causing underfitting, while a smaller lambda value leads to a denser noise matrix, which may give rise to overfitting concerns.

Note that since the L_0 norm makes the optimization in the second term challenging, the authors employed the L_1 norm to approximate it, as outlined in this work [4]. After substituting the L_1 norm into the equation, the final objective function is

$$\begin{aligned} O_{RobustNMF} &= ||X - UV - E||_F^2 + \lambda \sum_j [||E_{\cdot j}||_1]^2 \\ &= ||X - [U, I, -I] \begin{pmatrix} V \\ \begin{pmatrix} E^p \\ E^n \end{pmatrix} \end{pmatrix} ||_F^2 \\ &+ \lambda \sum_j [||E_{\cdot j}^p||_1 + ||E_{\cdot j}^n||_1]^2 \end{aligned} \tag{6}$$

3.2.2 Optimization Method

As the objective function is not convex with respect to the joint variables U, V, and E, the authors employ the multiplicative update rule [11] as the optimization method to iteratively update U, V, and E.

The update rule for U is:

$$U_{ij} = U_{ij} \frac{(\hat{X} V^T)_{ij}}{(UVV^T)_{ij}} \tag{7}$$

Given U, the update rule for V is:

Let $\widetilde{V} = \begin{pmatrix} V \\ \begin{pmatrix} E^p \\ E^n \end{pmatrix} \end{pmatrix}$.

$$\widetilde{V}_{ij} = max(0, \widetilde{V}_{ij} - \frac{\widetilde{V}_{ij} (\widetilde{U^T} \widetilde{U} \widetilde{V})_{ij}}{(S\widetilde{V})_{ij}} + \frac{\widetilde{V}_{ij} (\widetilde{U^T} \widetilde{X})_{ij}}{(S\widetilde{V}_{ij})}) \tag{8}$$

where $\widetilde{X} = \begin{pmatrix} X \\ 0_{1\times n} \end{pmatrix}$, $\widetilde{U} = \begin{pmatrix} U, I - I \\ 0_{1\times k} \sqrt{\lambda}_{e_1 \times m} \sqrt{\lambda}_{e_1 \times m} \end{pmatrix}$ and S is defined as

$$S_{ij} = |(\widetilde{U^T} \widetilde{U})_{ij}| \tag{9}$$

Finally, the update rule for E is:

$$E = E^p - E^n \tag{10}$$

where $E^p = \frac{|E| + E}{2}$, $E^n = \frac{|E| - E}{2}$, and $E^p \geq 0$, $E^n \geq 0$.

3.3 Noise Generation

3.3.1 Pepper and Salt Noise

In this paper [15], the authors adopt the pepper and salt noise for the following reasons:

(1) In reality, noise does not usually follow the Gaussian distribution.
(2) Techniques relying on least square estimation tend to be susceptible to this type of noise [3].

Figure 1: the ORL dataset contaminated by pepper and salt noise

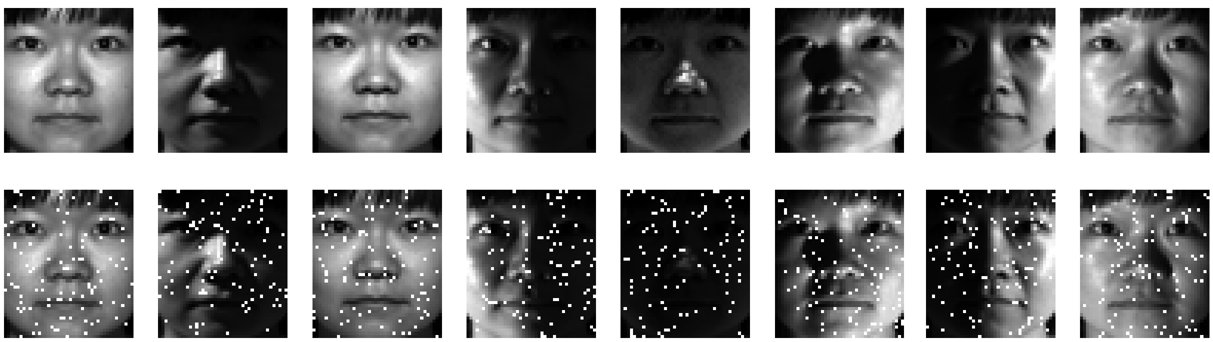


Figure 2: the YaleB dataset contaminated by pepper and salt noise

As a result, we randomly introduced 5% pepper and salt noise onto the image during experimentation without prior knowledge of the noise location.

3.3.2 Block Noise

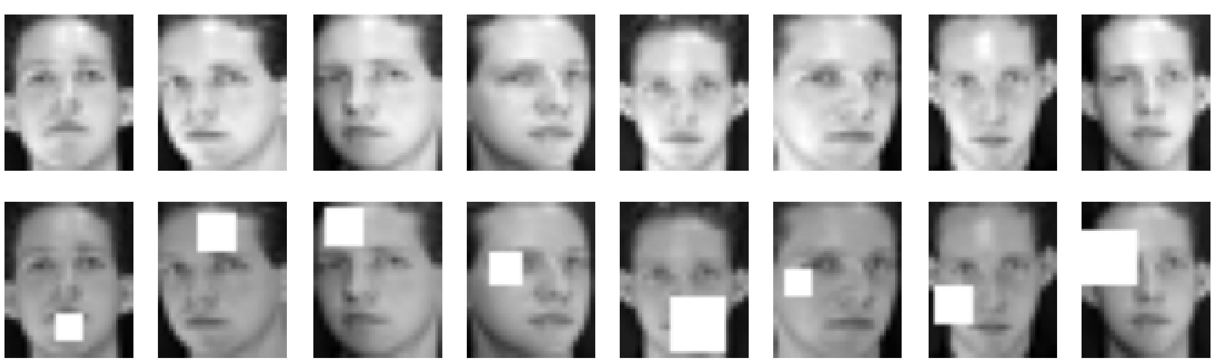


Figure 3: the ORL dataset contaminated by Block Noise

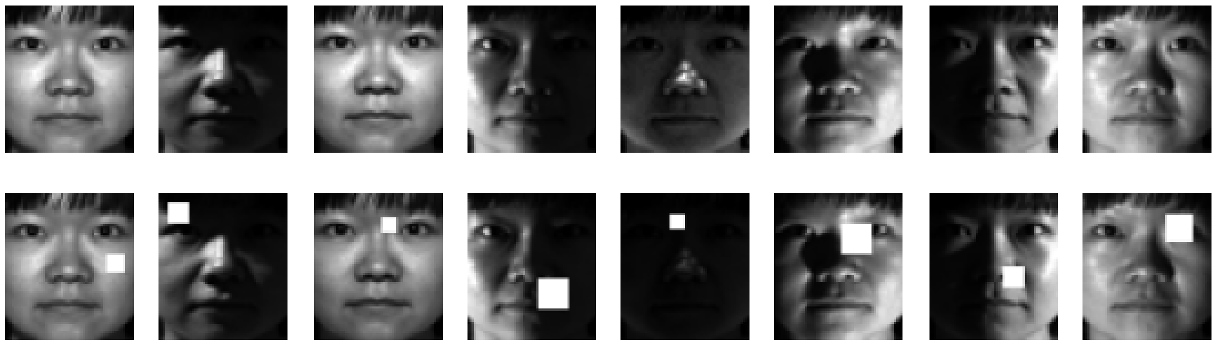


Figure 4: the YaleB dataset contaminated by Block Noise

As shown in 3 and 4, Block Noise are randomly positioned squares in random sizes from 4×4 to 8×8. Each image will be contaminated by one white block. Intuitively, these Block Noise will be larger outliers if it has larger size, or the corresponding area is darker.

3.3.3 Noise Comparison

Although Block Noise seems having a more regular shape than Pepper and Salt noise, the blocks contaminate a larger continuous area, which are larger outliers and would be more difficult to remove.

It is expected that the both $L_{2,1} - Norm$ NMF and L_1-Norm Regularised Robust NMF should generally perform better on Pepper and Salt Noise contaminated data than Block Noise contaminated data, as inpainting of additional large block areas are harder than small pixel noise generated by Pepper and Salt Noise.

4 Experiments

4.1 Evaluation Metrics

4.1.1 Root Mean Square Error

Root Mean Squared Error (RMSE) is used to measure the reconstruction error. For the vanilla NMF, dictionary and representations are learned to approximate the clean data $\hat{X}$. In the L_1-Norm Regularised Robust NMF, the UV also approximates the clean data. Therefore, the RMSE is the same as for vanilla NMF. The RMSE is defined as $\frac{1}{N}||X - UV||_F^2$, where N is the number of samples.

4.1.2 Accuracy and Normalized Mutual Information

The accuracy is calculated by comparing the predicted cluster labels with the true labels, while Normalized Mutual Information (NMI) is used to assess the quality of clustering results by quantifying the information shared between the true labels and the predicted cluster labels. K-Means clustering is employed to assign labels to each image based on the coefficients learned from NMF.

4.2 Setup

4.2.1 Runtime

The code is tested with Python 3.10.12 in Colab standard runtime and Python 3.9.16 in local Macbook M2 environment, respectively.

The minimal memory required to run the code is 2G. No GPU acceleration technique is applied. However, it could benefits from multiple CPU cores as the algoritms involve intensive matrix operations which can run simultaneously on multiple cores.

4.2.2 Datasets

We reduce the complexity of both given datasets ORL and YaleB, by setting the scale factor to 4, which makes the figure size one fourth of the original size.

4.2.3 Hyper-parameters

We set the maximal iteration times to 300, sample ratio to 0.9, random seed to 5328, sampling number to 5 and number of components to 100.

The reason we set the maximal iteration times to 300 is that the fitting process is very time-consuming while we have limited computational resource and time budget. Note that in some experiments the NMF models have not converged yet after 300 iterations.

4.3 Experiments on ORL

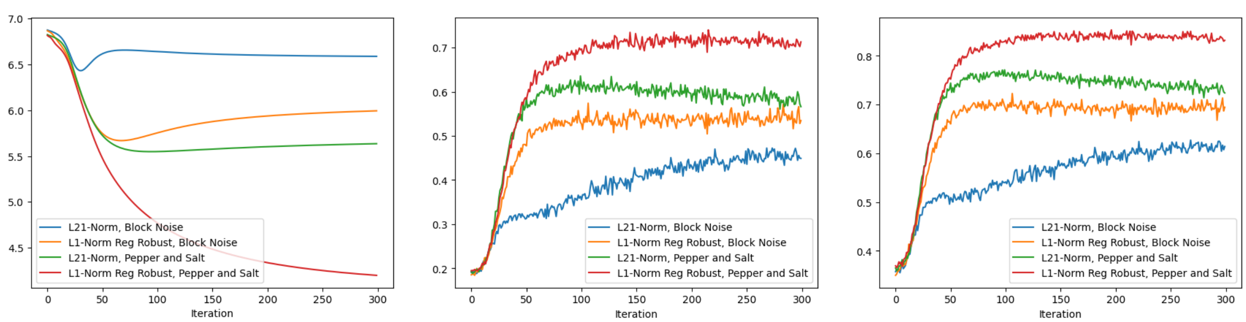
We carried out four experiments using three algorithms, i.e., L_2-Norm, $L_2, 1$-Norm and L_1-Norm regularised robust NMF, on five subsets randomly sampled from the ORL dataset contaminated by two types of noises respectively.

Algo	Noise	Avg. RMSE	RMSE Std	Avg. Acc	Acc Std	Avg. NMI	NMI Std
L2,1	Block	6.574	0.011	0.518	0.017	0.662	0.006
L1 Reg	Block	* 5.572	0.015	*0.663	0.010	*0.788	0.011
L2,1	Salt	5.558	0.018	0.651	0.015	0.782	0.009
L1 Reg	Salt	*4.317	0.078	*0.772	0.013	*0.868	0.005

Table 1: Evaluation metrics on the ORL dataset

4.3.1 Evaluation between $L_2, 1$-Norm and L_1-Norm regularised robust NMF

As you can see in 1, both $L_2, 1$-Norm NMF and L_1-Norm regularised robust NMF have stable performance over different subsets. The standard deviations are very small in comparison with the evaluation metric values.



(a) Avg. RMSE

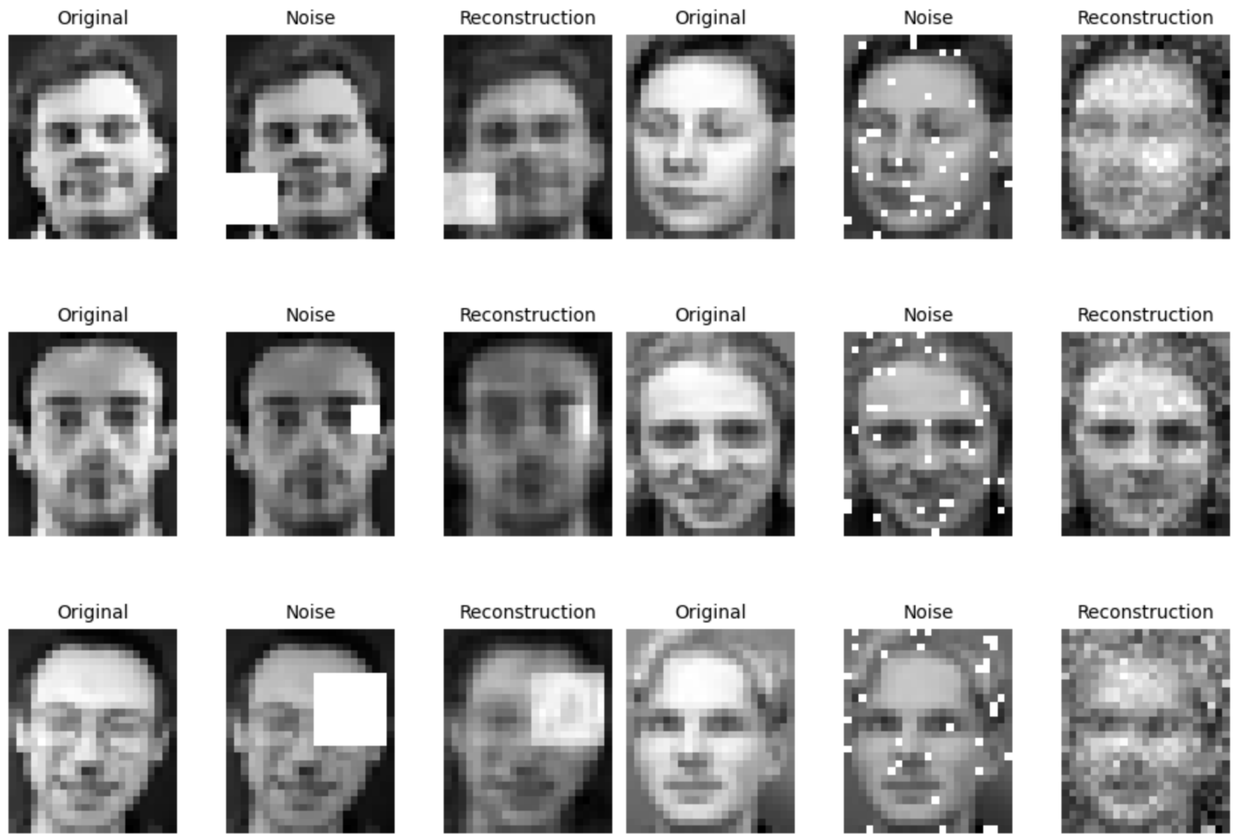
(b) Avg. Accuracy

(c) Avg. NMI

Figure 5: Evaluation metric comparison of $L_{2,1}$-Norm, L_1-Norm Regularised Robust, L_2-Norm NMF over the ORL dataset

5 clearly shows that all three metric trends of L_1-Norm Regularised Robust NMF and $L_{2,1}$-Norm NMF on both types of noise data are well separated. L_1-Norm Regularised Robust NMF has smaller RMSE, higher accuracy and NMI than $L_{2,1}$-Norm NMF, which indicates that L_1-Norm regularised robust NMF is more robust to the noise and the reconstructed images by L_1-Norm regularised robust NMF are closer to the original uncontaminated images.

Because L_1-Norm Regularised Robust NMF has much better evaluation metric values, the visual differences of the reconstructed images of $L_{2,1}$-Norm NMF and L_1-Norm Regularised Robust NMF are actually observable, as shown in 6 and 7. The images reconstructed by L_1-Norm Regularised Robust NMF look much better than the images reconstructed by $L_{2,1}$-Norm NMF.



(a) Block Noise

(b) Pepper and Salt Noise

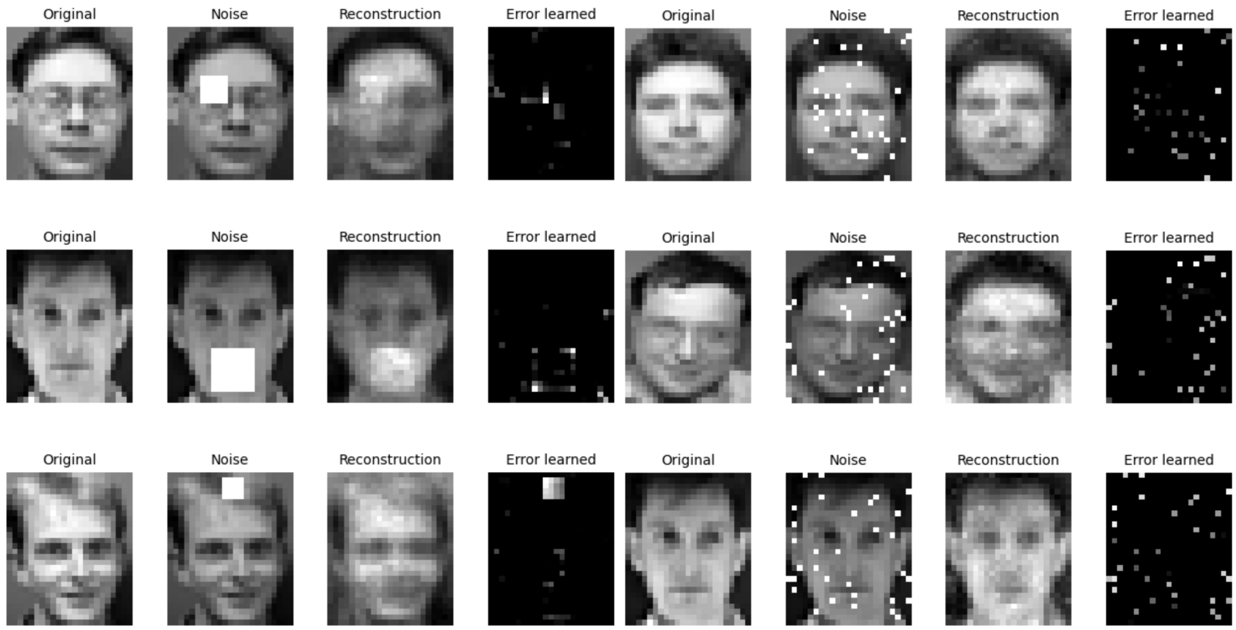
Figure 6: $L_{2,1}$-Norm reconstructed images of the ORL dataset

4.3.2 Evaluation between Block Noise and Pepper & Salt Noise

In terms of the differences of the performance on different noised data, 8 clearly shows that both $L_{2,1}$-Norm NMF and L_1-Norm Regularised Robust NMF have much better performance on Pepper and Salt Noise contaminated data.

Moreover, it takes much more iterations to reach the optimal solution on Block Noise contaminated data. Both algorithms start to perform worse after about 100 iterations on Pepper and Salt Noise contaminated data, while after 300 iterations, both algorithms have not converged yet on Block Noise data. These experimental results show that both algorithms are more robust to Pepper and Salt Noise than Block Noise, which is consist with that expected explained in previous noise description section.

Different from $L_{2,1}$-Norm NMF, L_1-Norm Regularised Robust NMF has an extra error term. In 7, we can see how the learned error term looks like. The noise learned on Pepper and Salt noise contaminated data looks very close to the original noise, while Block Noise learned are not that good.



(a) Block Noise (b) Pepper and Salt Noise

Figure 7: L_1-Norm Regularised reconstructed images of the ORL dataset

4.4 Experiments on YaleB

4.4.1 Evaluation Metric Results

Similar experiments are carried out on YaleB dataset contaminated by two types of noises, respectively. 2 shows the overall evaluation metric results. Generally, the differences of performance of $L_{2,1}$-Norm NMF, L_1-Norm Regularised Robust NMF on YaleB dataset are consistent with the results on Orl dataset.

Algo	Noise	Avg. RMSE	RMSE Std	Avg. Acc	Acc Std	Avg. NMI	NMI Std
L2,1	Block	5.413	0.008	0.221	0.014	0.279	0.018
L1 Reg	Block	4.592	0.014	0.268	0.010	0.322	0.011
L2,1	Salt	4.361	0.009	0.295	0.004	0.359	0.008
L1 Reg	Salt	4.030	0.013	0.299	0.005	0.351	0.010

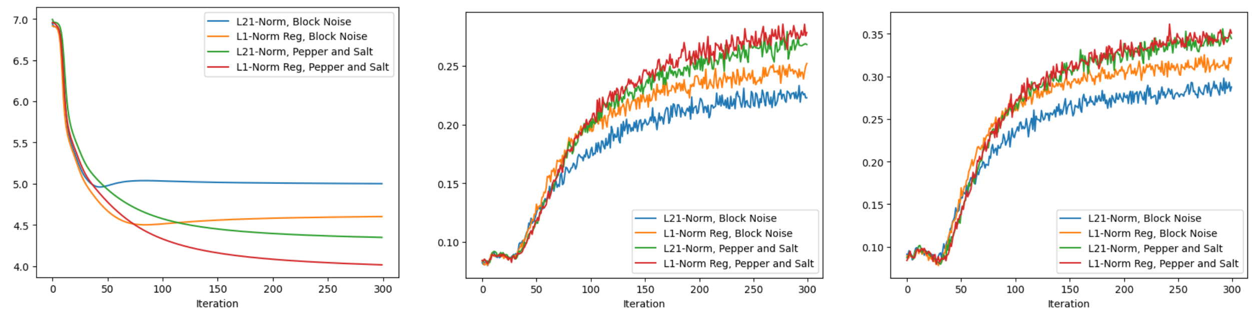
Table 2: Evaluation metrics on the ORL dataset

4.4.2 Performance Comparison between Orl and YaleB

However, as shown in 8, the differences of the evaluation metric values of $L_{2,1}$-Norm NMF and L_1-Norm Regularised Robust NMF on YaleB are smaller than that on Orl. Especially on the Pepper and Salt noise contaminated dataset, they almost have the same performance. This could be caused by the λ coefficient we used 0.02 is not optimal on this dataset, as that will be further discussed in next section.

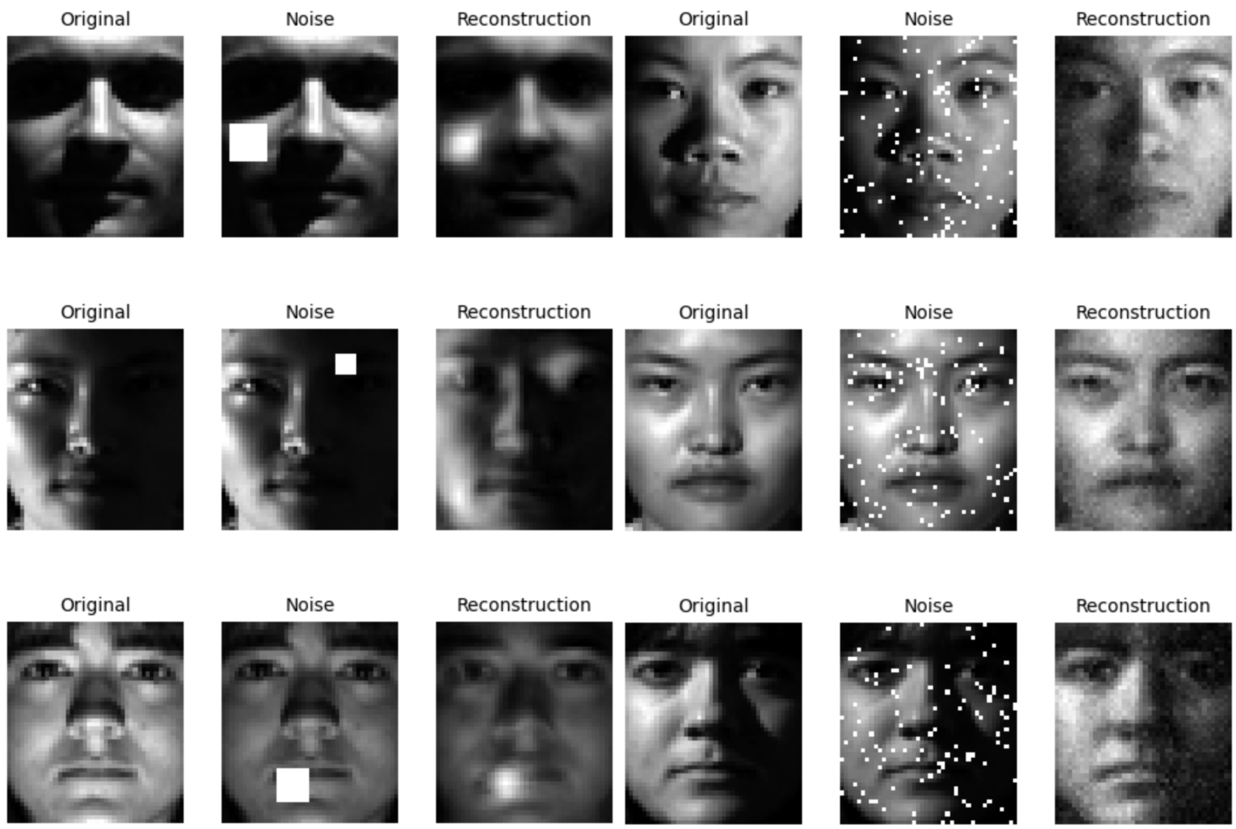
Meanwhile, the images of YaleB have strong contrast areas as shown in 9 and 10, which could be complete dark or very bright areas.

The error noise learned by L_1-Norm Regularised Robust NMF on YaleB is generally quite good. It also show that extremely large outlier pixels on YaleB are easier to learn. For example, the white block noise positioned in the complete dark area of the original image is exactly learned as shown in 10



(a) Avg. RMSE (b) Avg. Accuracy (c) Avg. NMI

Figure 8: Evaluation metric comparison of $L_{2,1}$-Norm, L_1-Norm Regularised Robust over YaleB dataset



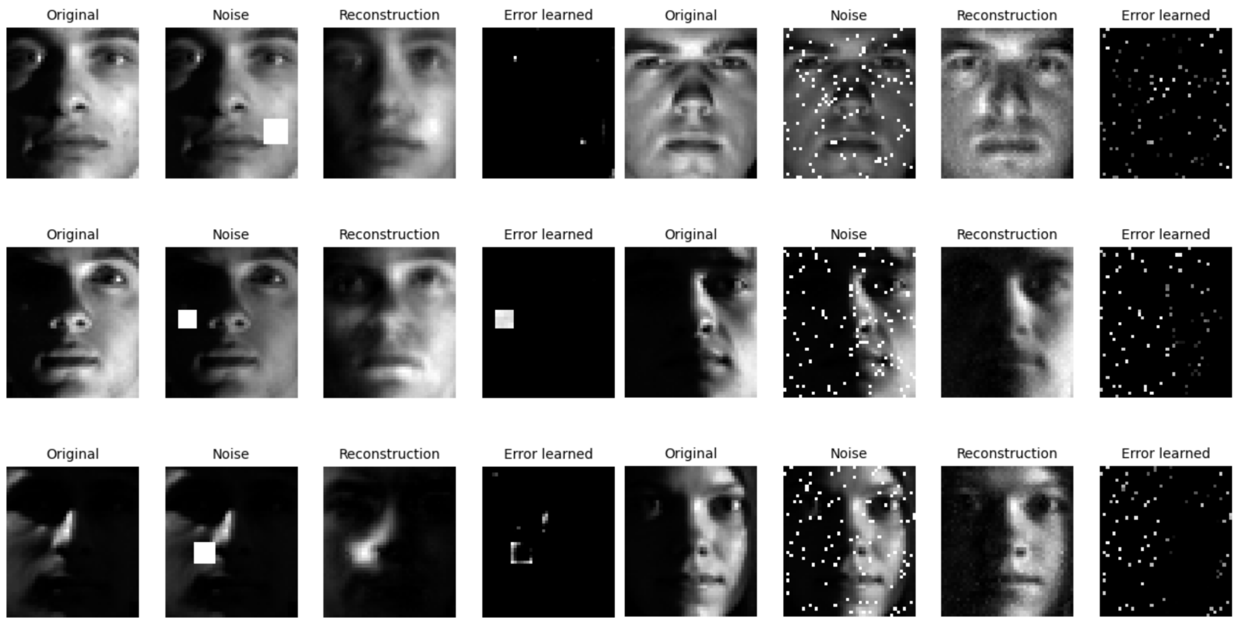
(a) Block Noise (b) Pepper and Salt Noise

Figure 9: $L_{2,1}$-Norm reconstructed images of YaleB dataset

4.5 Experiment different λ coefficients

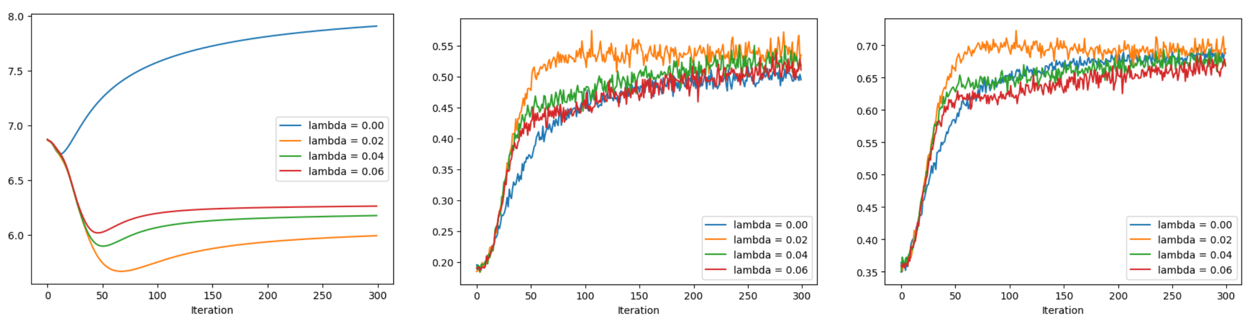
In addition, we carried out four experiments to observe the differences between the performance of L_1-Norm Regularised Robust NMF with different λ coefficients, 0, 0.02, 0.04 and 0.08. From 11, we can see that λ 0.02 has the best evaluation metric results.

Note that this optimal λ is only observed on Orl contaminated by Block Noise. Theoretically, it could yield different optimal results on different datasets contaminated by different noise.



(a) Block Noise (b) Pepper and Salt Noise

Figure 10: L_1-Norm Regularised reconstructed images of YaleB dataset



(a) Avg. RMSE (b) Avg. Accuracy (c) Avg. NMI

Figure 11: Evaluation metric comparison of L_1-Norm Regularised Robust with different λ

Due to fitting the model is very time-consuming, we couldn't carry out further experiments to obtain the optimal λ coefficients for all experiments. For the other experiments, we set λ coefficients to the same value 0.02.

5 Conclusion

In all the experiments carried out on ORL dataset and YaleB dataset contaminated by Block Noise and Pepper and Salt Noise, both algorithms $L_{2,1}$-Norm NMF and L_1-Norm Regularised Robust NMF have stable performance with very small deviations.

L_1-Norm Regularised Robust NMF shows the power of error learning. It has significantly better performance than $L_{2,1}$-Norm NMF in almost every experiment. Although $L_{2,1}$-Norm NMF reduces the impact of outliers by not squaring the residuals, it is not as adaptive as L_1-Norm Regularised Robust NMF to noises due to the lack of ability of learning errors.

Specifically, L_1-Norm Regularised Robust NMF learns Pepper and Salt noise better than Block noise, which is consistent to that explained previously, L_1-Norm Regularised Robust NMF is designed to be more robust to randomly unknown position of additive noise.

We also found that the λ coefficient of L_1-Norm Regularised Robust NMF is significant to the robustness of noise. However, the optimal λ coefficient varies on different noise levels. To achieve the best performance, the λ coefficient should be carefully chosen as it cannot be optimised by L_1-Norm Regularised Robust NMF algorithm itself.

References

[1] Anil M Cheriyadat and Richard J Radke. Non-negative matrix factorization of partial track data for motion segmentation. In *2009 ieee 12th international conference on computer vision*, pages 865–872. IEEE, 2009.

[2] M CHU, F DIELE, R PLEMMONS, and S RAGNI. Optimality, computation, and interpretation of nonnegative matrix factorizations (version: October 18, 2004).

[3] Fernando De la Torre and Michael J Black. Robust principal component analysis for computer vision. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 1, pages 362–369. IEEE, 2001.

[4] David L Donoho. For most large underdetermined systems of equations, the minimal 1-norm near-solution approximates the sparsest near-solution. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 59(7):907–934, 2006.

[5] Paul Fogel, S Stanley Young, Douglas M Hawkins, and Nathalie Ledirac. Inferential, robust non-negative matrix factorization analysis of microarray data. *Bioinformatics*, 23(1):44–49, 2007.

[6] David Guillamet and Jordi Vitria. Non-negative matrix factorization for face recognition. In *Catalonian Conference on Artificial Intelligence*, pages 336–344. Springer, 2002.

[7] A Ben Hamza and David J Brady. Reconstruction of reflectance spectra using robust nonnegative matrix factorization. *IEEE Transactions on Signal Processing*, 54(9):3637–3642, 2006.

[8] Patrik O Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of machine learning research*, 5(9), 2004.

[9] Hyunsoo Kim and Haesun Park. Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis. *Bioinformatics*, 23(12):1495–1502, 2007.

[10] Deguang Kong, Chris Ding, and Heng Huang. Robust nonnegative matrix factorization using l21-norm. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 673–682, 2011.

[11] Daniel Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. *Advances in neural information processing systems*, 13, 2000.

[12] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[13] Li Liu, Douglas M Hawkins, Sujoy Ghosh, and S Stanley Young. Robust singular value decomposition analysis of microarray data. *Proceedings of the National Academy of Sciences*, 100(23):13167–13172, 2003.

[14] Christophe Samson, Laure Blanc-Féraud, Gilles Aubert, and Josiane Zerubia. A variational model for image classification and restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(5):460–472, 2000.

[15] Bin Shen, Bao-Di Liu, Qifan Wang, and Rongrong Ji. Robust nonnegative matrix factorization via l 1 norm regularization by multiplicative updating rules. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 5282–5286. IEEE, 2014.

[16] Yi Wu, Bin Shen, and Haibin Ling. Visual tracking via online nonnegative matrix factorization. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(3):374–383, 2013.