Assignment 1

Tutors
Zhuo Huang, Yuhao Wu

Group members

Rajiv Mehta	rmeh0608	540771859
Edward Ji	ziji4098	510477226
Yao Ke	yake9601	510459679
Daniel Kovalenko	dkov2101	500499357

Abstract

Non-negative matrix factorisation (NMF) is a proven way to compress data into an interpretable dictionary matrix and a compressed sparse code matrix, allowing for the compression of data, identifying common patterns and further downstream clustering classifications. However, in real world applications such as facial recognition, the image of a subject may often be obstructed. As such, we tested the error robustness of two types of NMF algorithms, L_1-Norm based NMF and Frobenius-Norm based NMF, to see if they are able to handle Gaussian, block and grid noise. This noise replicates obstructions and variations present in real images. We found that both NMF algorithms performed similarly, handling Gaussian noise whilst struggling to handle occlusions such as those simulated by block and grid noise. This is likely due to the large change in overall brightness of the image in our occlusion tests and the destruction of the overall structure of the face which the NMF dictionary is learning. In the future, an occlusion of varied brightness as well as varied occlusion sizing can quantify the exact impact of these factors.

1 Introduction

Non-negative Matrix Factorisation (NMF) is a mathematical technique to take an arbitrary $m \times n$ matrix X, and decompose it into two smaller matrices, D and R. This dimension reduction technique is used to retain the important information in X to decrease the computational expense of classification tasks.

$$X \approx DR$$

where:

- $X \in \mathbb{R}^{m \times n}$ is the original matrix (with m rows and n columns),
- $D \in \mathbb{R}^{m \times k}$ is a matrix representing latent factors,
- $R \in \mathbb{R}^{k \times n}$ is a matrix of coefficients,
- k is the number of latent factors or features.

Reducing the amount information in the dataset through NMF can lead to a reduction in accuracy due to the models inability to determine the optimal number of features or clusters or guarantee convergence to a local minimum, where small changes in the parameters can cause the global optimisation to change significantly[1]. The advantages of NMF such as the reduction of space for processing, the sparsity, non-negativity maintaining the original properties of the A and being easy to interpret has led to NMF to being a widely popular machine learning modelling technique used in image recognition and text classification. With the reduction of data it is important to test the robustness of these algorithms to understand how accurate NMF algorithms are in classification.

This report assesses two NMF algorithms with two distinct objective functions, being L_1 norm and *Frobenius norm*. Their performance is tested against datasets containing labelled images or human faces. The robustness of the program is determined by the reconstruction of these images with different types of noises that are represented in the real world. The metrics involved are reconstruction error and the accuracy of the downstream tasks on the reconstructed images, as well as normalised mutual information.

2 Literature Review

Before starting the design and implementation of these NMF algorithms, it is first important to see how previous iterations of these algorithms have performed in the past.

'Robustness of NMF Algorithms Under Different Noises' by Kang et.al., covers this very topic. In their research they used a 'salt and pepper noise and random matrix noise filter to create noise in two different datasets, ORL and Extended Yale B [2]. From here they implemented a *Standard NMF Algorithm* and $L_{2,1}$*-Norm Based NMF Algorithm* to test the robustness's of NMF. In their experiment, the same environmental setup was used for both algorithms with a maximum number of iterations being set to 500 and having the critical event stream processing (ESP) value to be set to $1e-5$, causing the iterations to terminate if the value is less then ESP. In this experiment they found compared with salt and pepper noise, the random matrix noise has a minor impact on the clustering test of the two NMF algorithms. [2] However, since the random matrix noise will modify all the pixels, it will cause a more significant relative reconstruction error. Comparing the algorithms it was also found that the standard NMF algorithm and $L_{2,1}$-Norm Based NMF Algorithm had a very similar level of robustness with the later having a slight advantage over both datasets and noise filtering methods.

Similar to the previous study, *'Analyze the robustness of three NMF algorithms (Robust NMF with L_1 norm,L_{2-1} norm NMF, L_2 NMF)'* by Zeng et.al. covers three different NMF algorithms on the ORL and YaleB datasets. [3] In this experiment the 'salt and pepper' and 'block-occlusion' noise were used. To test the robustness they first pre-processed the data by normalization to ensure that all values were between 0-1. Further for the YaleB dataset they resized the images to $42x48$ pixels and the ORL dataset to be $30x37$ pixels. They then compared the L_1 robust regularized NMF,L2-1 norm NMF, L2 NMF algorithms through three different evaluation metrics, 'Root Mean Square Error' (RMSE), 'Average Accuracy' and 'Normalized Mutual Information (NMI)'. To optimize the performance, 90% of the data was used to train the models where it was repeated 5 times to obtain average evaluation metrics. From here, they then tested over a series of components K=10, 20, 30, 40 to further optimize the models. From these results they determined that all the NMF algorithms tested were robust though L_2 performs best when using the large YaleB dataset and when using a smaller dataset ORL, $L_{2,1}$ is best. [3]

'Analysis of the robustness of NMF algorithms' by Diaz et.al. follows the previous two studies but uses a *Gaussian Noise* and *Laplace Noise* filter with the *Salt and Pepper noise* filter to test the robustness of L_1 norm,L_{2-1} norm, L_2-norm NMF algorithms on the ORL and YaleB dataset [4]. In their study, they used three parameters to test the robustness and the feature predictions, *Relative Reconstruction Errors (RRE)* which describes the similarity between the reconstructed matrix and original clean matrix, *Average Accuracy* and *NMI*. Experimenting the 'feature selection' they found that when using a Gaussian noise filter on the ORL dataset, that the three algorithms produced a similar level of performance with the number of components being between 40-60 producing the best results. The experiment on robustness yielded similar results as the found that all three algorithms were robust on smaller dataset sizes achieving the best result on Gaussian Noise filtering. Though when applying the Salt and Pepper noise filter, the accuracy dropped by 40% with performance of L_1 and L_{2-1} being similar [4].

In the paper *'Robust dual-graph discriminative NMF for data classification'* by Lu et.al, the authors proposed a supervised NMF algorithm to consider the differentiable of different data samples called *Robust Dual-graph Discriminative Non-negative Matrix Factorization (RDGDNMF)* which uses a

$L_{1,1/2}$ norm to minimise the loss function of the NMF[5]. In their study they tested the robustness and accuracy of 18 different algorithms ranging from an unsupervised NMF to a supervised deep NMF with a Support Vector Machine used for classification. They tested this on four different datasets; 'MINISt', 'Caltech 101', 'COIL20', and 'ORL' with a 'Poisson' or 'Gaussian' noise filter. From their experiments they found that the robustness of their proposed RDGDNMF algorithm performed better compared to the other algorithms, though when more noise was added, the model became less stable on both noise filters. Comparing their proposed method with the 'Constrained NMF', the proposed method performed significantly better, with a close to 15% average increase between the NMF algorithms.

Another study *'Another Robust NMF: Rethinking the Hyperbolic Tangent Function and Locality Constraint'* by Shen et.al. aimed to try and boost robustness of NMF algorithms through the addition of a hyperbolic tangent function creating a TanhNMF [6]. In this experiment, they compared a CauchyNMF, CINMNF, L_{21}NMF, tanhNMF and a tanhNMF$^+$ (tanNMF with a localtiy constraint) on four different face datasets; 'Yale','YaleB','UMIST' and 'PIE' datasets. To evaluate these algorithms, they ran each algorithm ten times where the average results of the evaluation metrics; 'clustering accuracy' and 'normalized mutual information' were compared. From there study, it can be seen that the tanhNMF$^+$ appeared to perform the best on average across all datasets with the standard NMF performing significantly worse. To further prove this, the authors conducted a t-test to compare the results and determined that tanhNMF+ is superior to the baseline methods including CIMNMF, CauchyNMF,K-means, L21NMF, NMF and tanhNMF. But on UMIST dataset, the p-value between tanhNMF and tanhNMF+ is over 0.05, and thus they are thought to be statistically identical indicating that the hyperbolic transformation increased the robustness of NMF [6].

In the paper *'Noise-tolerant clustering via joint doubly stochastic matrix regularization and dual sparse coding'* by Shi.et.al the authors proposed a stochastic matrix regularization and dual sparse coding framework (DSNMF) to overcome the weaknesses of standard NMF algorithms such as the ability to handle outliers, graph constructions being fixed and not-adaptive and no sparse constraints both on the basis matrix and the coefficient matrix, which may cause important information to be ignored during clustering [7]. In this study to test the robustness, the CMU PIE dataset was used where 'Block Occlusion' and 'Salt and Pepper' noise was added to corrupt the databases which was then evaluated using Clustering Accuracy (ACC), Normalized Mutual Information (NMI) and Purity (PUR). In their experiment, they compared a k-means clustering algorithm, NMF, $L_{2,1}$-NMF and Correntropy (replacing Eucledian distance on DSNMF). When looking at block occlusion, all algorithms significantly reduced in overall accuracy when the block size gradually increased in size, with Correntropy being slightly better across the board. Comparing the DSNMF, GNMF and CGNMF with a standard NMF on the Salt and Pepper affected dataset, it can be seen that the three modified NMFs performed on average 10% better with then the standard NMF.

3 Methodology

As seen in previous literature, common tests for the robustness of NMF algorithms involved the usage of the *ORL* and *YaleB* datasets with various types of noise, *Gaussian, Block* and *Grid Noise*. For this study, we will be focusing on L_1 and *Frobenius* (L_2) *norm* algorithms to determine the robustness of standard NMF algorithms. To evaluate the robustness, *Relative Reconstruction Error (RRE), Average Accuracy (ACC)* and *Normalized Mutual Information (NMI)* will be used to compare the performance of the NMF algorithms.

3.1 ORL and YaleB Dataset

The ORL dataset consists of 40 subjects, where each subject has 10 photos. The images had slight variations in lighting as well as facial orientation and features such as glasses vs no glasses. The raw dataset pictures all had a 92 by 112 pixel dimension. To reduce the computational complexity of the matrix operations, we processed all the images from this dataset to a down scaled 46 by 56 dimension.

The YaleB dataset consists of 38 subjects each with 64 images. There is a much larger variation in lighting and facial orientation in this dataset. All images are 168 by 192 pixels. We down sampled the images to 42 by 48 pixels to reduce computational complexity. In this dataset lighting is a lot more varied but there are no additional features such as glasses.

3.2 L_1 Norm

Given a matrix where each column corresponds to one sample, the L_1 norm loss is defined as the mean L_1 norm of residuals of the reconstructed images. Formally, suppose there is a dataset $X \in \mathbb{R}_+^{m\times n}$ where m is the number of features , and n is the number of samples, the residual of a reconstruction using $D \in \mathbb{R}_+^{m\times r}$ and $R \in \mathbb{R}_+^{r\times n}$ is given by $E = X - DR$. The L_1 norm loss is the mean of L_1 norm of the columns of E,

$$L = \frac{1}{n}\sum_{j=1}^{n}\|E_j\| = \frac{1}{n}\sum_{j=1}^{n}\sum_{i=1}^{m}|e_{ij}|.$$

Then, the derivative of loss with respect to each entry in the residual matrix is

$$\frac{\partial L}{\partial e_{ij}} = \frac{1}{n}\,\mathrm{sign}(e_{ij}). \tag{1}$$

In the case of ORL and YaleB image datasets, we flatten each image in $\mathbb{R}_+^{h\times w}$ into a single vector in $\mathbb{R}_+^{m}$, where $m = hw$ the number of pixels in an image.

Similar to more traditional methods using Frobenius norm, the algorithm for L_1 norm updates D and R in turns by solving for the derivative of L with respect to each while holding the other constant. First, suppose R is constant, using the multivariate chain rule, the derivative of L with respect to any entry d_{ij} in the dictionary is $\frac{\partial L}{\partial d_{ij}} = \sum_k\sum_l \frac{\partial L}{\partial e_{lk}}\frac{\partial e_{lk}}{\partial d_{ij}}$. Notice that $e_{lk}(d_{ij})$ is constant when $l \neq i$, therefore, the derivative can be reduced to

$$\frac{\partial L}{\partial d_{ij}} = \sum_k \frac{\partial L}{\partial e_{ik}}\frac{\partial e_{ik}}{\partial d_{ij}}. \tag{2}$$

Since $e_{ik} = x_{ik} - d_i \cdot R_k = x_{ik} - \sum_l d_{il}r_{lk}$, the derivative of r_{ik} with respect to d_{ij} is the coefficient of w_{il} when $l = j$ in the sum.

$$\frac{\partial r_{ik}}{\partial d_{ij}} = -h_{jk}. \tag{3}$$

Substituting 1 and 3 into 2 yields, $\frac{\partial L}{\partial d_{ij}} = \sum_k -\frac{1}{n}\,\mathrm{sign}(e_{ik})r_{jk} = -\frac{1}{n}r_j \cdot \mathrm{sign}(e_i)$. In matrix form, the gradient can be more compactly written as

$$\frac{\partial L}{D} = -\frac{1}{n}\,\mathrm{sign}(E)R^T.$$

where $\mathrm{sign}(E)$ denotes the element-wise sign of E.

By symmetry, the derivative of L with respect to the representation matrix R is

$$\frac{\partial L}{R} = -\frac{1}{n}D^T\,\mathrm{sign}(E).$$

Using gradient descent with learning rate η, an additive update rule for NMF with L_1 loss is as follows,

$$D_{t+1} = D_t + \frac{\eta}{n}\,\mathrm{sign}(E)R_t^T$$
$$R_{t+1} = R_t + \frac{\eta}{n}D_{t+1}^T\,\mathrm{sign}(E)$$

where D_t and R_t are D and R at the t-th iteration respectively.

Critically, the rules do not guarantee that the matrices will be positive even if they start off positive ($D^0 \in \mathcal{D}$ and $R^0 \in \mathcal{R}$). Therefore, we clip the entries of both matrices to $[0, \infty)$ at the end of each iteration.

Mathematically speaking, the L_1 norm is not as sensitive to outliers as many other measures of error, which may make it perform better in some applications.

3.3 L$_2$ Norm - Frobenius Norm

Traditionally, NMF uses the squared Frobenius norm of residual as its objective function. The task is to find two non-negative matrices that minimize the loss function. Formally, the loss function is $L = \|X - DR\|_F^2$ where $\mathcal{D} = \mathbb{R}_+^{m\times r}, \mathcal{R} = \mathbb{R}_+^{r\times n}$.

From the lectures, if we differentiate and carefully choose the learning rate in gradient decent, there exists a nice multiplicative rule for optimizing the factorization as follows,

$$R_{ij}^{t+1} = R_{ij}^t \frac{(D^T X)_{ij}}{({D^t}^T D^t R^t)_{ij}} \qquad D_{ij}^{t+1} = D_{ij}^t \frac{(X R^{t+1})_{ij}}{(D^t R^{t+1} {R^{t+1}}^T)_{ij}}.$$

It's more convenient than the additive rule for L$_1$ norm in the sense that it guarantees the non-negativity of both matrices at all times, because the property is preseved under matrix and scalar multiplications.

3.4 Noise

Let $Z \in \mathbb{R}^{h\times w}$ be the noise matrix we add element-wise to each image $X \in \mathbb{R}_+^{h\times w}$, where h and w are the height and width of the image respectively. There are three types of noise defined below and demonstrated in Figure 1.

Gaussian noise. Noise that follows a normal distribution. $Z_{ij} \sim \mathcal{N}(\mu, \sigma^2)$ where the parameters default to $\mu = 0$ and $\sigma = 25$.

Block noise. A randomly positioned white square block. $Z_{ij} = 255 \times 1_{\{(i,j)\in\mathcal{B}\}}$, where $\mathcal{B} = [I, I + k] \times [J, J + k]$ is a square block of size k. The pair (I, J) is a random variable that represents the top left corner of $\mathcal{B}$. The indexes I and J are randomly sampled from discrete uniform distributions $\mathcal{U}\{1, h - k\}$ and $\mathcal{U}\{1, w - k\}$ respectively. The parameter k defaults to 0.3 minimum side length of the original image and controls the side length of the block.

Grid noise. A grid of white square blocks that are regularly spaced and randomly offset from the top left corner. There are two control parameters: k for the size of the blocks in grid, and l for the gap between the blocks. The noise can be seen as a repetition of the $(k + l) \times (k + l)$ matrix

$$255 \begin{bmatrix} I_k & 0_{k\times l} \\ 0_{l\times k} & 0_{l\times l} \end{bmatrix}.$$

The noise is then a tiling of $\left(\lceil \frac{h}{k+l} \rceil + 1\right)$ rows and $\left(\lceil \frac{w}{k+l} \rceil + 1\right)$ columns of such matrices that would cover the image, allowing an offset of up to but not including $(k+l)$ both vertically and horizontally. The offset (I, J) is randomly chosen where $I, J \sim \mathcal{U}\{0, k + l - 1\}$.

Each type of noise is applied to each pixel by addition and constrained to $[0, 255]$, that is, $\hat{X} = \min\{\max\{X + Z, 0\}, 255\}$ where $\hat{X}$ is the processed image.

All three types of noise occur naturally in the real world. Due to the physical limitations of modern instruments, Gaussian noise is present on almost all photographs taken, especially when there's not enough lighting. By amplifying the Gaussian noise on the ORL and YaleB datasets across the image, one can test the robustness of an algorithm against images taken in rougher conditions. Block noise can represent an unexpected obstruction due to some other object coming into the scene or simply damaged data. Grid noise takes it further by dividing the block noise in to some chunks scattered across the image, making it potentially harder for the algorithm to learn the whole picture.

3.5 Metrics

To compare the performance of different NMF algorithms under different types of noise, we employ three metrics.

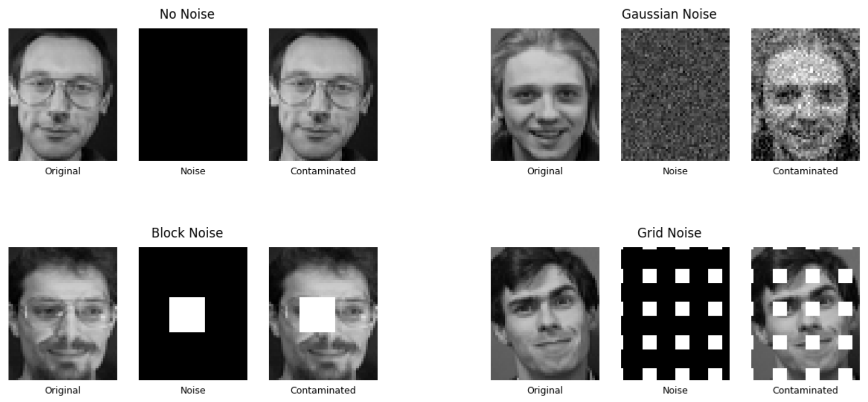


Figure 1: Examples of different types of noise applied to an image.

Relative reconstruction error (RRE). Let $\hat{X}$ denote the contaminated dataset with added noise, and X denote the original dataset. Let D and R denote the learned factorization on the contaminated dataset $\hat{X}$. The RRE is defined as:

$$\text{RRE} = \frac{\|X - DR\|_F}{\|X\|_F}.$$

Since NMF is usually a preprocessing step to perform other more costly algorithms, one of which is clustering, the performance of the aforementioned algorithms is also evaluated on the downstream task of k-means clustering. This is possible because the datasets contain the true labels $y \in \mathbb{N}^n$ on the images. Given that $\hat{y}$ are the clustering predictions after NMF, the final two metrics are defined as follows.

Accuracy (ACC).

$$\text{ACC}(y, \hat{y}) = \frac{1}{n}\sum_{i=1}^{n} 1_{\{y_i = \hat{y}_i\}}.$$

Normalized mutual information (NMI).

$$\text{NMI}(y, \hat{y}) = \frac{2I(y, \hat{y})}{H(y) + H(\hat{y})}$$

where I is the mutual information function and H is the entropy function.

3.6 10-fold

To ensure the results are reliable, the evaluation is performed using a typical 10-fold setup. The final result is taken from the average of 10 trials performed on the training split of each fold. The standard deviation of all metrics are also recorded to conclude the stability of the algorithms. This is roughly equivalent to taking `90%` random data for 10 trials but more reliable since we make sure all data is used 9 times.

4 Experiment and Results

4.1 Results

4.1.1 L$_1$ Norm

With regards to the ORL dataset, the L$_1$ norm had the highest accuracy when factorising the data without any noise applied to it, followed by Gaussian noise, block noise, and finally grid noise. A similar trend was observed with the YaleB dataset.

			RRE		ACC		NMI	
			mean	std	mean	std	mean	std
Loss	Dataset	Noise						
L_1 Norm	ORL	None	0.214	0.00815	0.734	0.01973	0.857	0.00820
		Gaussian	0.160	0.00086	0.694	0.03343	0.824	0.01931
		Block	0.337	0.00864	0.382	0.02648	0.553	0.02259
		Grid	0.578	0.01883	0.188	0.01327	0.350	0.01802
	YaleB	None	0.205	0.00097	0.227	0.01432	0.293	0.01423
		Gaussian	0.204	0.00064	0.227	0.00985	0.305	0.01225
		Block	0.583	0.00463	0.104	0.00605	0.112	0.00462
		Grid	0.684	0.00690	0.091	0.00462	0.088	0.01092
F Norm	ORL	None	0.124	0.00033	0.711	0.02815	0.840	0.01978
		Gaussian	0.144	0.00039	0.700	0.02621	0.831	0.01571
		Block	0.323	0.00348	0.309	0.01400	0.481	0.01822
		Grid	0.551	0.00257	0.186	0.00874	0.349	0.01501
	YaleB	None	0.184	0.00061	0.240	0.00950	0.326	0.01191
		Gaussian	0.193	0.00056	0.233	0.01475	0.318	0.01730
		Block	0.627	0.00264	0.099	0.00400	0.099	0.00530
		Grid	0.761	0.00274	0.084	0.00327	0.076	0.00428

Figure 2: Main experiment results.

The L_1 norm algorithm had a much higher accuracy in general when working with the ORL dataset than the YaleB dataset, and also had a wider range of averages from 0.734 in the no noise data, to 0.188 in the grid noise data. Meanwhile, the YaleB dataset's averages were in a smaller range, from 0.227 with no noise having the highest accuracy, to 0.091 using grid noise having the lowest accuracy. This is reflected by the difference in the standard deviations of the two datasets - the standard deviations of the experiments using the ORL dataset ranged from 0.03343 (using Gaussian noise) to 0.01327 (using grid noise). Meanwhile, the YaleB dataset's standard deviations only ranged from 0.01432 with no noise, to 0.00462 with grid noise.
The Relative Reconstruction Error (RRE) of the L_1 norm largely follows the inverse of the trends identified above with the accuracy. For the ORL dataset, it increases from 0.214 with no noise applied, to a maximum of 0.578 with grid noise applied. Meanwhile for the YaleB dataset, the RRE remains within a tighter range of 0.124 with no noise applied, to a maximum of 0.684 with block noise applied.
The normalised mutual information (NMI) also largely follows the trends identified with the accuracy above. The ORL dataset had a wide range of NMI values from 0.857 with no noise applied being the highest, and 0.350 with grid noise applied being the lowest. The YaleB dataset was once again much more tightly constrained, with a maximum of 0.305 with Gaussian noise applied, and a minimum of 0.088 with grid noise applied.

4.1.2 Frobenius Norm

The Frobenius norm follows many of the patterns identified above with the L_1 norm. It has a higher accuracy overall across the four noise types with the ORL dataset, albeit with a wider range of accuracy values, than the YaleB dataset. The highest accuracy value that the Frobenius norm recorded using the ORL dataset was 0.711 with no noise applied, while the lowest recorded was 0.186 when grid noise was applied. Meanwhile, when using the YaleB dataset, the Frobenius norm's highest average accuracy was 0.240 with no noise applied, and its lowest average accuracy was 0.084 with grid noise applied.
As identified with the L_1 norm, the Frobenius norm sees a much higher standard deviation in its error values when training on the ORL dataset compared to the YaleB dataset. The standard deviation values for no noise, Gaussian noise, block noise, and grid noise were 0.02815, 0.02621, 0.01400 and 0.00874 respectively. Meanwhile, the YaleB dataset had standard deviations of 0.00950 for no noise, 0.01475 for Gaussian noise, 0.00400 for block noise, and 0.00327 for grid noise, making them smaller than their ORL counterparts.
The RRE of the Frobenius norm, like the L_1 norm, follows the inverse of the accuracy trends. The

RRE increases as the accuracy values decrease, with the lowest RRE starting at 0.124 with no noise for the ORL dataset, and increasing to 0.551 with grid noise. Similarly, the RRE increases from a low of 0.184 with no noise for the YaleB dataset, to a high of 0.761 with grid noise.
The normalised mutual information also follows the accuracy trends across both datasets. For the ORL dataset, it starts at a high of 0.840 with no noise applied, and decreases with each noise type applied down to a low of 0.349 with grid noise applied. For the YaleB dataset, it decreases from a high of 0.326 with no noise applied, to a low of 0.076 with grid noise applied. Of note are the significantly lower NMI values in the YaleB dataset compared to the ORL dataset.

4.2 Discussion

The above results show that on average, the L_1 norm performed better than the Frobenius norm in terms of accuracy when undertaking non-negative matrix factorisation, as shown by the higher accuracy values in the L_1 norm compared to the Frobenius norm. The exceptions to this were the Gaussian noise experiment using the ORL dataset (0.694 vs 0.700), the no noise experiment using the YaleB dataset (0.227 vs 0.240), and the Gaussian noise experiment using the YaleB dataset (0.227 vs 0.233). This potentially could be due to the fact that the Frobenius norm is more sensitive to outliers than the L_1 norm [8], as the squaring of the high error values that outliers cause leads to very high adjustment values in the optimisation functions which do not reflect the adjustment values that should be applied from non-outliers. This in turn leads to the L_1 norm having a higher level of accuracy in general.
Of note as well is the difference in standard deviation in the accuracies observed using both algorithms between the ORL dataset and the YaleB dataset. The L_1 norm showed an average standard deviation of 0.02323 across the 4 experiments with the ORL dataset, while the YaleB dataset had an average standard deviation of 0.00871 across the 4 experiments. Similarly, the Frobenius norm showed an average standard deviation of 0.01928 in the 4 experiments with the ORL dataset, while the 4 YaleB dataset experiments had an average standard deviation of 0.00788. This shows that for the L_1 norm, the average standard deviation was 1.67 times greater for the ORL dataset than the YaleB dataset, while for the Forbenius norm the average standard deviation was 1.44 times greater for the ORL dataset than the YaleB dataset.
This also ties in to the average accuracies that the two algorithms had overall between the two datasets. The L_1 norm had an average accuracy of 0.4995 across the 4 ORL experiments, and an average accuracy of 0.1623 for the YaleB dataset. The Frobenius norm meanwhile had an average accuracy of 0.4765 for the ORL dataset experiments, while it only achieved an average accuracy of 0.164 for the YaleB dataset experiments. This indicates that the algorithms performed significantly better in terms of accuracy on the ORL dataset than they did on the YaleB dataset.
A potential explanation for this variation in the accuracies and the standard deviations could be due to the nature of the data itself. The main difference in the images of the ORL dataset is the angle at which the photos are taken - some images have subjects facing directly ahead, some slightly angled left or right, etc. Meanwhile, the main difference in the images of the YaleB dataset is the lighting - in some images, the subjects are clearly lit, in other the light source is above or below the subject, in some images the subject is barely lit, etc. A few examples are shown below:

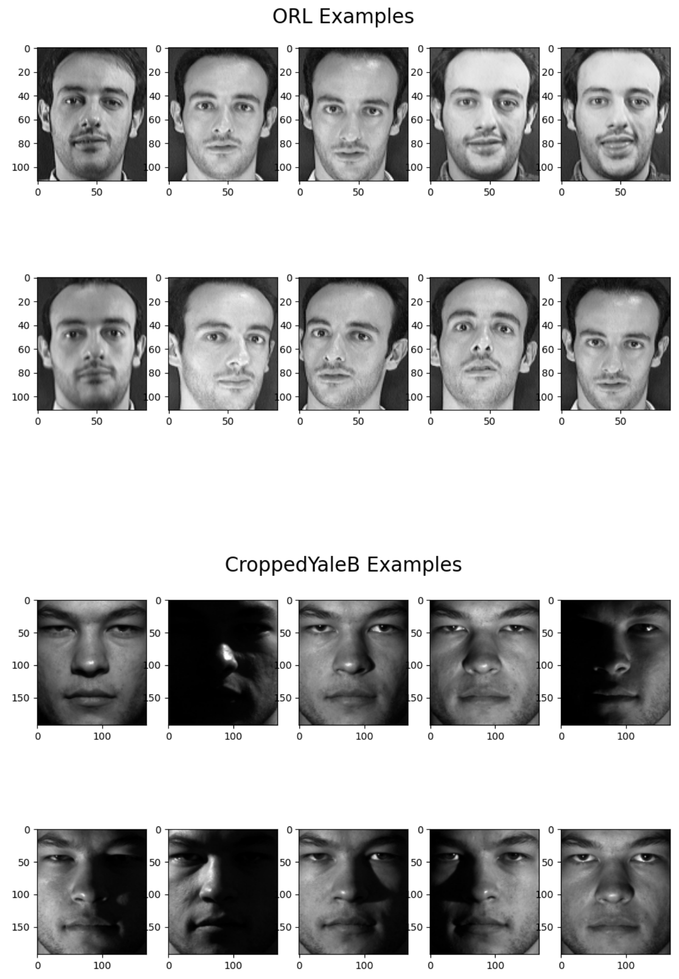


Figure 3: Different examples from the ORL and YaleB datasets.

The accuracy values indicate that both the L_1 norm and the Frobenius norm struggle more with poor lighting than with faces being angled at different directions, as displayed by the higher accuracy for the ORL dataset compared to the YaleB dataset. However, the variation in standard deviations for the ORL dataset may indicate that both the models are learning the locations of certain facial features, rather than the facial features themselves. For example, a model may learn that a feature such as a nose should be in the center of the image, and may score a high accuracy for images that have the subject and hence the nose centered, but when the subject has their face and hence their nose tilted to the left or right, the model may score a lower accuracy as it is expecting the nose to be in the center of the image. Meanwhile, because the YaleB examples all have the subject facing directly ahead, the features are all relatively in the same location, meaning the error values that the models generate would be less varied than those in the ORL experiments, leading to a lower standard deviation in the error values. This suggests that the L_1 and Frobenius norms are not robust to variations in the angles of the images.

The above results also suggest that the L_1 and Frobenius norms are not very robust to block and grid noise. The most obvious piece of evidence for this is the lower accuracy values of the block and grid experiments compared to the no noise and Gaussian noise experiments across both datasets and across both norms. However, another piece of evidence that supports this is the lower overall accuracy values of the YaleB dataset. As mentioned above, the two norms both seem to struggle with poor lighting, as that is the main variation between the YaleB images. Poor lighting can be considered a form of noise - just as a solid block or grid of noise could obscure part of a face and impact the ability of a model to learn features in that area, so too can a shadow darken part of a face so much that features are indistinguishable, and thus similarly impact a model's ability to learn features. An example is provided below, with an example YaleB image with no noise on the left, and the same image with block noise applied on the right:

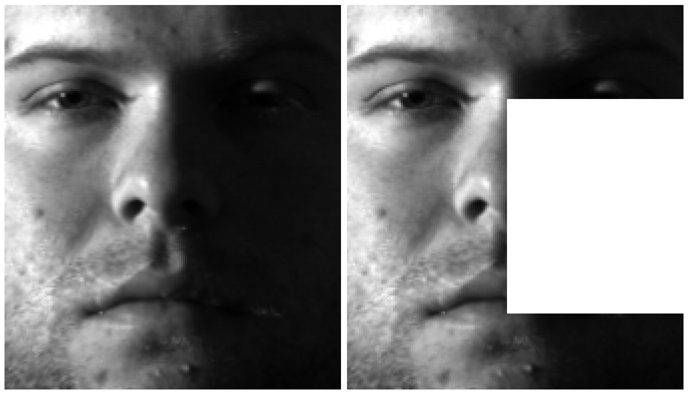


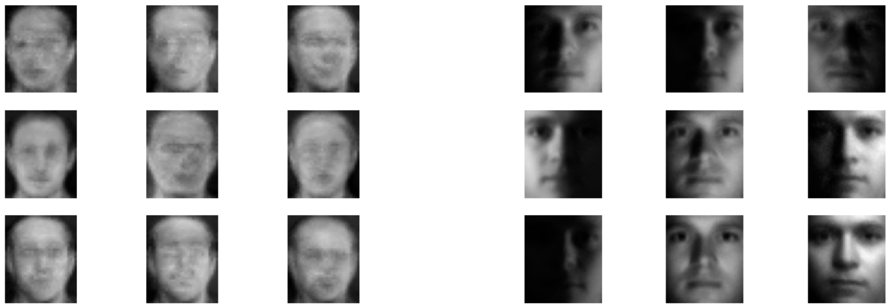
Figure 4: A YaleB image both with and without block noise applied.

The shadow in the left image plays a similar role to the noise in the right image - they both obscure facial features, making it difficult for the algorithms to determine what features should be there. The no noise experiments using the YaleB dataset still contained this ”shadow” noise, as it was part of the original images, and could therefore have contributed to the two models' poor performance in the YaleB dataset.

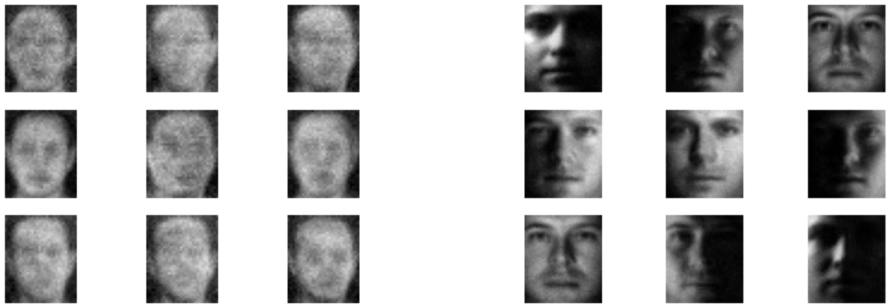
Below are a few visual examples to show the results of the reconstructed matrix using the L_1 norm, with the ORL dataset on the left and the YaleB dataset on the right:

Figure 5: $\mathbf{L}_1$ **Norm Reconstruction Examples**

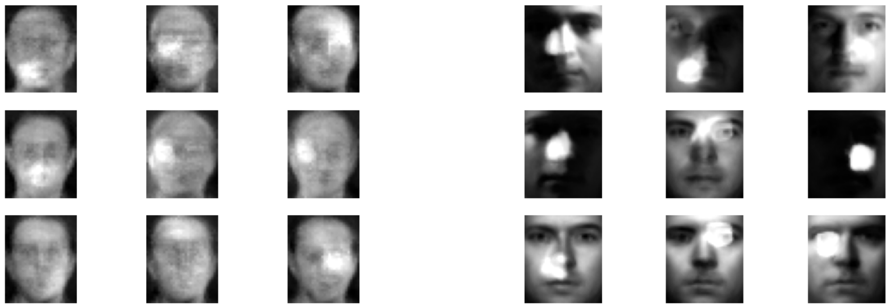
No Noise



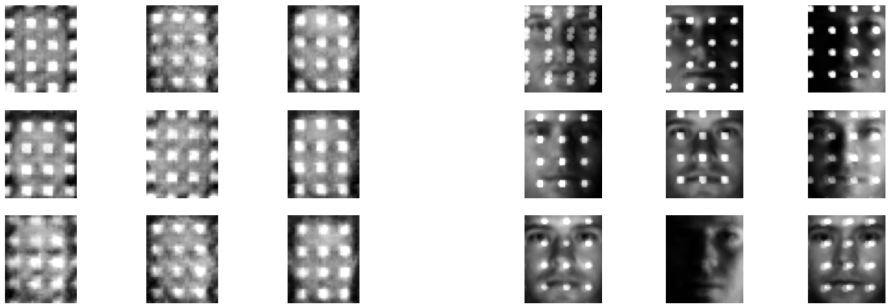
Gaussian Noise



Block Noise



Grid Noise

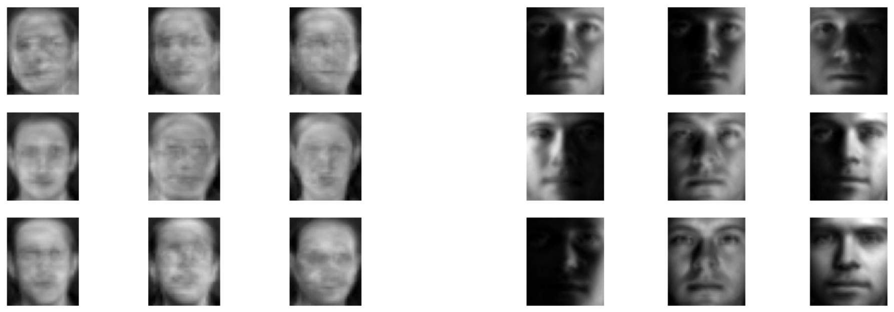


The above example images show that despite the low accuracy values, the L_1 norm seems to perform better at reconstructing the YaleB images rather than the ORL images. The ORL images are much more blurry and difficult to find features in, whereas the YaleB images are much clearer. This supports the idea that the model is learning the positions of features rather than the features themselves. Across both datasets, it is clear that the L_1 norm is still learning some noise, as seen by the white marks in the block noise and grid noise images, or the grainy nature of the Gaussian noise images. However, the L_1 norm is still much more robust in countering this noise compared to the Frobenius norm. This can be seen for example in the block noise images, where the size of the noise blocks is much smaller in the L_1 norm compared to the Frobenius norm, and in the case of the YaleB images, the L_1 norm is able to more clearly display the features usually covered by the noise, as shown for example in the middle, bottom middle, and bottom left images in the block noise results for the YaleB images. Similarly, with the grid noise results, the size of the grid squares is smaller than those in the Frobenius norm's results, once again indicating that the L_1 norm is better able to learn the characteristics of the images than the Frobenius norm, and is thus more robust to different types of noise. The L_1 norm also seems to be resistant to Gaussian noise, as the reconstructed images from the YaleB dataset largely look similar to the original images, just more grainy. The same cannot be said for the ORL images, as they look very unclear even with no noise, however the size of the noise in both the block noise and grid noise experiments is smaller and less pronounced than in their Frobenius counterparts, once again suggesting that the L_1 norm is more robust to noise than the Frobenius norm.

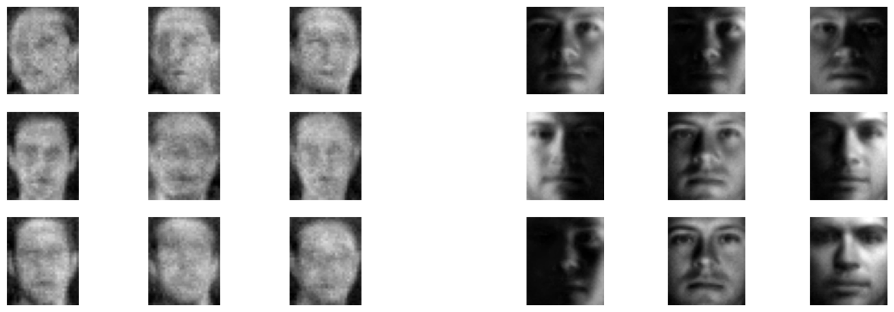
The following are a few visual examples to show the results of the reconstructed matrix using the Frobenius norm, with the ORL dataset on the left and the YaleB dataset on the right:

Figure 6: **Frobenius Norm Reconstruction Examples**

No Noise

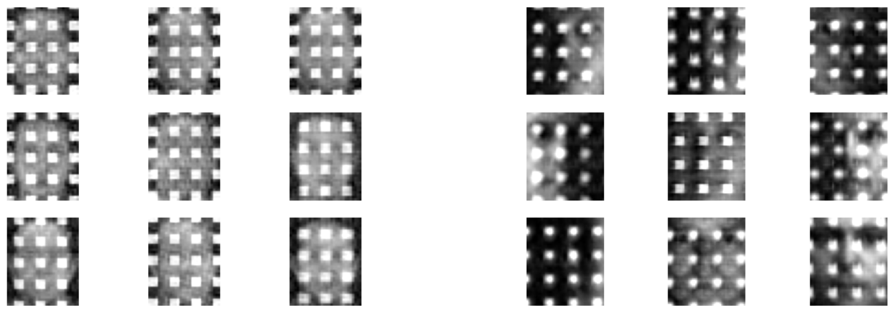


Gaussian Noise



Block Noise

Grid Noise



In contrast to the higher accuracy values in the results, the Frobenius norm's reconstruction of the ORL images is not very clear. Across all four noise types, the reconstructed images look blurry, as if the subjects were moving when the images were being taken. This supports the theory that the model is learning the position of certain features, rather than the features themselves, as indicated by the standard deviation of the accuracy results. For the images where the subject is at an angle (for example, the top left images of each set), the model seems to be trying to reconstruct the subject both in the center of the image, and in the location in the original image. Meanwhile, for the images where the subject is looking directly ahead (such as the middle left images), the image seems to be less blurry, as the model's reconstructed position of the different facial features matches the location of the features in the original images. This gets progressively worse with each noise type applied - by the time block noise and grid noise is applied, much of the detail in the original features is lost, and there is little indication that the model is actually learning the facial features themselves (as shown by the lack of detail in the sections with noise), unlike in the YaleB dataset where there is some indication of facial features being learnt. Despite the low accuracy values, the Frobenius norm seems to reconstruct the YaleB images quite well, even with Gaussian noise applied. The effect of Gaussian noise seems to simply lower the quality of the reconstructed images, making them blurrier and less defined. However, the facial features of the images are still generally retained, indicating that the model does learn facial features despite this Gaussian noise. With the block noise, it seems to learn the positions of the noise block in the reconstructed image, but there are some hints of features being learnt. For example, in the image examples given above, the top left image has the outline of an eye in the middle of the noise block, and the bottom right as well as center right images both have faint outlines of lips, despite the mouth areas being covered by noise. The areas covered by shadows however do not resemble any facial features, giving credence to the idea that the shadows in the images also act as noise, imparing the ability of the model to learn features. With regards to grid noise, the model does not seem to be able to completely replace the noisy areas with facial features, however the size of the grid squares seems to be smaller than the original grid noise, and certainly smaller than the grid squares than those seen in the ORL dataset, indicating that the Frobenius norm is able to at least somewhat learn facial features in order to fill in some of the space taken up by the noise.

In terms of the RRE and NMI values, for both norms they largely follow the expected trends relating to accuracy. As accuracy increases, the RRE should decreases, as the RRE is a measure of the error a model produces. Meanwhile, as accuracy increases, so should NMI, since NMI measures how similar two sets of clustered labels are. If the predicted labels have a similar proportion of each unique label to the true labels and it can overall provide a good approximation of the true labels, NMI will be higher. NMI is a very similar measure to accuracy that also takes into account if any erroneous label assignments are more ”forgivable” due to class imbalance and a higher raw probability of that particular predicted label in the dataset. If these more acceptable errors are encountered, the NMI is penalised to a lesser degree, reflecting the lower importance of these errors. As such, for the ORL and YaleB datasets, NMI does not provide significant improvement over accuracy as the classes are all balanced. These trends are observed in both the norms tested across both the datasets, with RRE being inversely proportional to accuracy, and NMI being proportional to accuracy.

Overall, these results are quite contradictory. On the one hand, the numerical results such as the accuracy and NMI values indicate that both the L_1 norm and Frobenius norm are able to confidently predict the class of an image from their facial reconstruction in the ORL dataset more so than the YaleB dataset, supporting the idea that the models are better at learning the features of the ORL dataset compared to the YaleB dataset. Yet the visual examples show the opposite, with the ORL reconstructions being blurry and the features indistinguishable, while the YaleB reconstructions are quite clear. The numerical values also hide how the models are able to reconstruct certain facial features to a moderate degree despite noise being present. An explanation for this discrepancy could once again be the different angles of the ORL images - since the objective images have facial features across several different locations in different images, the model may be learning that reconstructing features in all of those locations is acceptable. For example, because the ORL images could have a subject's chin at the bottom left, right and middle of a set of example images, the model may attempt to minimise the loss function by calculating values that cause the resulting reconstructed matrix to have pixels in all those locations, when in reality it should only have them in one location depending on what the original image was.

5 Reflection

Overall, our experiment was performed to a good standard, and generally followed the scientific method in determining the robustness of the L_1 and Frobenius NMF algorithms both with and without noise. However, there were some issues that occurred. The first major issue was the stability of the L_1-norm. The L_1-norm was very unstable when training, as can be seen in figure 9 in the appendix. This led to us needing to undertake early stopping, which potentially limited the ability of the L_1-norm to properly learn all the detail in the images. Additionally, when testing the block and grid noise, the noise applied was one solid white colour, which was just the maximum colour value possible. This meant that our noise was not as reflective of real noise as it could have been, as in reality the noise encountered in images is hardly the brightest or darkest possible colour. Instead, real noise tends to have some similarity to its surroundings in terms of shading and brightness (e.g. an object covering a face will tend to have the same brightness as the face if it is in a similar location). To make future experiments more accurate to real noise, a variety of colour or grayscale values should be used for the added noise.

6 Conclusion and Future Work

In this study, we explored the robustness of NMF algorithms by adding *Gaussian Noise*, *Block Noise* and *Grid Noise* to the *ORL* and *YaleB* dataset on the L_1 and *Forbenius* NMF datasets. The study showed that on average the L_1 norm performed better then the Frobenius norm when it came to classifying images on both datasets, with the models performing better on the ORL dataset. Though, this was contradictory when compared to the visual reconstruction of the models, as it was clear that the YaleB dataset produced clearer features/images. The NMF algorithms also appeared to be quite susceptible to different angles in the images and though they can handle small amounts of noise (as seen with Gaussian noise), the block and grid noise tended to lower the accuracy and overall robustness of the algorithms.

In order to gain a better understanding of what exactly these models are learning, future experiments should incorporate a more basic control dataset that does not vary facial angles, lighting, or many other features, in order to have a starting point from which to compare datasets such as the ORL and YaleB datasets. Additionally, more varied datasets should also be included. A combination of different facial angles and lighting setups in the same dataset would be beneficial to identifying further capabilities of these two algorithms. More benefits could also be gained from having datasets where the differences in the angles of the subjects in the images is more pronounced - this way, it could be determined whether the models are trying to recreate the original matrix through placing the features in all possible positions, leading to the distorted results that were seen with the ORL dataset. Another benefit could be gained from having more variations of noise - creating noise of different shapes (such as circles, triangles, etc) as well as of different brightness levels and not just the maximum colour value as was done for the grid and block experiments would help determine more about how robust the two norms are to noise.

Aspect	**Gaussian Noise (MLE)**	**Laplacian Noise (MLE)**
Log-Likelihood Equation	$\ell(\theta) = -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i - h_\theta(X_i))^2$	$\ell(\theta) = -n\log(2b) - \frac{1}{b}\sum_{i=1}^{n}\lvert x_i - h_\theta(X_i)\rvert$
Objective Function	$\theta^* = \arg\min_\theta \frac{1}{n}\sum_{i=1}^{n}(x_i - h_\theta(X_i))^2$	$\theta^* = \arg\min_\theta \frac{1}{n}\sum_{i=1}^{n}\lvert x_i - h_\theta(X_i)\rvert$
Positives	- Captures squared deviations, ensuring smooth optimization. - Closed-form solutions available for linear models. - Computationally efficient due to differentiability.	- Robust to outliers due to the absolute error formulation. - Better suited for datasets with heavy-tailed noise distributions.
Negatives	- Sensitive to outliers since squared deviations magnify their influence. - Assumes Gaussian noise, which may not always hold.	- Optimization can be harder (nondifferentiable at 0). - No closed-form solutions for most models. - May converge more slowly in high-dimensional spaces.

References

[1] A. N. Langville, C. D. Meyer, R. Albright, J. Cox, and D. Duling, "Algorithms, initializations, and convergence for the nonnegative matrix factorization," 2014. [Online]. Available: https://arxiv.org/abs/1407.7299

[2] M. Kang, J. Zhao, and Z. Han, "Robustness of nmf algorithms under different noises," *EAI Endorsed Transactions on Internet of Things*, vol. 9, no. 1, p. e4, 2023.

[3] C. Zeng, J. Tian, and Y. Xu, "Analyze the robustness of three nmf algorithms (robust nmf with l1 norm, l2-1 norm nmf, l2 nmf)," *arXiv.org*, 2023.

[4] A. Díaz and D. Steele, "Analysis of the robustness of nmf algorithms," *arXiv.org*, 2021.

[5] G. Lu, C. Leng, B. Li, L. Jiao, and A. Basu, "Robust dual-graph discriminative nmf for data classification," *Knowledge-based systems*, vol. 268, pp. 110 465–, 2023.

[6] X. Shen, X. Zhang, L. Lan, Q. Liao, and Z. Luo, "Another robust nmf: Rethinking the hyperbolic tangent function and locality constraint," *IEEE access*, vol. 7, pp. 31 089–31 102, 2019.

[7] Z. Shi and J. Liu, "Noise-tolerant clustering via joint doubly stochastic matrix regularization and dual sparse coding," *Expert systems with applications*, vol. 222, pp. 119 814–, 2023.

[8] Q. Ke and T. Kanade, "Robust subspace computation using l1 norm," 2003.

Appendix

Notations

$\mathbb{R}_+$ denotes the set of non-negative real numbers.

Given an m by n matrix X,

- x_{ij} denotes its ij-entry,
- x_i denotes its ith row, and
- X_j denotes its jth column.

Given any vector x and y, $x \cdot y$ denotes their dot product. Scalar multiplication between a and b is simply written as ab.

Define the function $\mathsf{sign} : \mathbb{R} \to \{-1, 0, 1\}$ as

$$\text{sign}(x) = \begin{cases} -1 & \text{if } x < 0 \\ 0 & \text{if } x = 0 \\ 1 & \text{if } x > 0 \end{cases}$$

1_X is the indicator function of event X.

Extra Figures

Dataset	Learning Rate	RRE	ACC	NMI
ORL	0.01	0.168	0.708	0.847
ORL	0.02	0.146	**0.782**	**0.888**
ORL	0.05	**0.137**	0.757	0.881
ORL	0.1	1.46e+19	0.733	0.853
ORL	0.2	7.45e+97	0.025	0
YaleB	0.01	0.315	0.0862	0.0804
YaleB	0.02	0.256	0.15	0.173
YaleB	0.05	0.213	0.205	0.275
YaleB	0.1	**0.206**	**0.21**	**0.278**
YaleB	0.2	2.53e+20	0.094	0.108

Figure 7: Experiment results for tuning learning rate of NMF with L_1 norm. Bold figures indicate the best result within dataset, red figures indicate a clear divergence. A learning rate of around 0.05 is the best. For the YaleB dataset which has far more samples, with a learning rate any less than 0.05, the NMF with L_1 norm loss doesn't learn fast enough. Other the other hand, with a learning rate any higher than that, we risk having exploding gradients and a divergence, especially in a smaller dataset like ORL.

Loss	Dataset	#Components	RRE	ACC	NMI	Time (s)
L_1 Norm	ORL	10	0.191	0.632	0.791	4.065
	ORL	20	0.173	0.682	0.814	4.625
	ORL	50	0.159	**0.688**	**0.829**	6.794
	ORL	100	**0.156**	0.682	0.810	9.461
	ORL	200	0.160	0.575	0.735	8.498
	YaleB	10	0.261	0.152	0.174	60.924
	YaleB	20	0.230	0.183	0.261	69.331
	YaleB	50	0.194	0.231	0.298	89.537
	YaleB	100	0.176	**0.264**	**0.331**	98.179
	YaleB	200	**0.163**	0.252	0.313	111.232
F Norm	ORL	10	0.186	0.642	0.807	1.262
	ORL	20	0.163	**0.690**	**0.823**	1.023
	ORL	50	0.141	0.677	0.815	2.156
	ORL	100	**0.138**	0.685	0.809	4.023
	ORL	200	0.151	0.520	0.675	9.562
	YaleB	10	0.256	0.128	0.170	7.573
	YaleB	20	0.224	0.189	0.263	10.621
	YaleB	50	0.182	0.239	0.324	19.699
	YaleB	100	0.155	**0.290**	**0.375**	28.319
	YaleB	200	**0.138**	0.286	0.344	47.004

Figure 8: Experiment results for tuning the number of components. Bold figures indicate the best result obtained for combinations of loss and dataset. Using a value in the scale of 50 as the number of components is enough to yield moderate performance without taking too long to train, especially when the dataset is large like YaleB. Decreasing the dimension in the representation makes the model less prone to overfitting. Subsequent experiments and the main results use the number of classes in the dataset as the number of components, which is slightly less than 50.

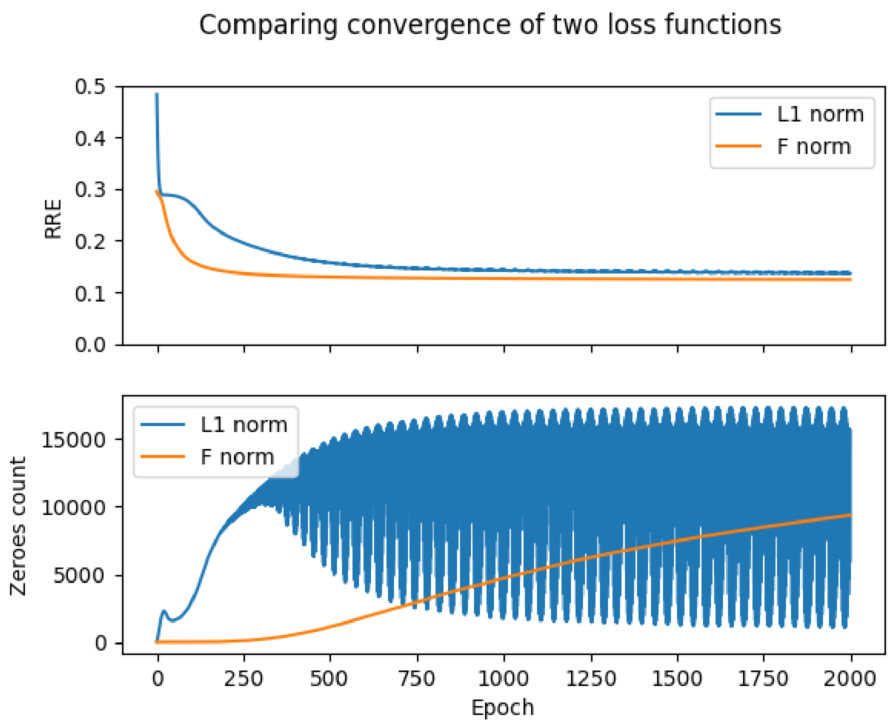


Figure 9: Plot of RRE and sparsity using the two loss functions over the number of epochs. Sparsity is measured by the total number of values close to 0 in both D and R. The plot shows that L_1 norm with additive updates converges much slower than the Frobenius norm with multiplicative updates and its sparsity is less stable. However, it reaches roughly the same level of performance and expected sparsity after 2000 epochs.

			RRE		ACC		NMI	
			mean	std	mean	std	mean	std
Loss	Dataset							
L_1 Norm	ORL	None	0.152	0.00098	0.718	0.01835	0.849	0.01099
		Gaussian	0.178	0.00096	0.665	0.02463	0.813	0.01117
		Block	0.266	0.00347	0.432	0.02149	0.605	0.01913
		Grid	0.517	0.00305	0.207	0.00923	0.374	0.01580
	YaleB	None	0.254	0.00261	0.146	0.01132	0.178	0.01208
		Gaussian	0.295	0.00146	0.086	0.00403	0.077	0.00514
		Block	0.293	0.00225	0.096	0.00365	0.097	0.00476
		Grid	0.295	0.00108	0.090	0.00524	0.088	0.00909
F Norm	ORL	None	0.127	0.00043	0.731	0.01142	0.852	0.00594
		Gaussian	0.148	0.00055	0.703	0.02892	0.829	0.01506
		Block	0.322	0.00259	0.282	0.01433	0.453	0.02329
		Grid	0.550	0.00228	0.184	0.00619	0.345	0.01585
	YaleB	None	0.188	0.00065	0.237	0.00971	0.319	0.01059
		Gaussian	0.199	0.00069	0.234	0.01265	0.316	0.01520
		Block	0.624	0.00216	0.097	0.00523	0.098	0.01188
		Grid	0.762	0.00321	0.083	0.00305	0.074	0.00274

Figure 10: Experiment results with regularization $\lambda = 0.1$. The result doesn't show enough improvement from the main experiment results in Figure 2 without regularization. Therefore, the evidence suggests against the use of regularization with the model and hyper-parameters in this report.

Contributions

Name	Contribution	Sections Contributed To
Rajiv Mehta	25%	Introduction, Literature Review, Editing/Formatting, Conclusion
Daniel Kovalenko	25%	Initial L1-Norm code and experiments, Results, Discussion, Conclusion, Editing
Edward Ji	25%	Refine implementation, Methodology section and Extra Figures in report
Yao Ke	25%	Code Experiments, Finalising Code Base, Editing Report, Writing Abstract

Table 1: Names and their contributions