Assignment 1

Tutors: (Jazlyn Lin, Vincent Qu)
Group members: (Qi Lu, SID: 490196302, unikey: qilu3337
Shiming Guo, SID: 500672949, unikey: sguo4070
Zheng Wang, SID: 480403605, unikey zwan6697)

Abstract

As an important matrix factorization model, non-negative matrix factorization(NMF) is widely used in the fields of data mining and machine learning. It is often used to extract low-dimensional, sparse, and meaningful features from a collection of non-negative data vectors[1]. This report uses 2 datasets, i.e., ORL dataset and extended YaleB dataset. For each dataset, four new datasets are created, two of them add the block-occlusion noise with different block size, the other two adds the salt pepper noise with different noise pixel proportion. The purpose is to enforce 3 NMF algorithms, i.e. L_2 norm NMF, $L_{2,1}$ norm NMF and CIM-NMF in reconstructing two type noise datasets in light and heavy noise settings. 3 evaluate metrics, i.e., Relative Reconstruction Errors (RRE), Average Accuracy and Normalised Mutual Information (NMI) are used to evaluate the robustness of each algorithm. Starting with the introduction, this section explains the main idea of NMF, the problem intended to solve and the importance of it, the related application of NMF and the overview of the NMF used in this assignment. Following that is the previous work section. Methodology section will explain the details of the noises and NMF algorithms, in this section the robustness of each algorithm is also explained based on the theoretical view. The next part is about the experiment, it introduces the datasets, experiment setup, experimental results and some personal reflects. After the experiment, the conclusion part summarises the results, the experiment limitations and discusses the future work.

1 Introduction

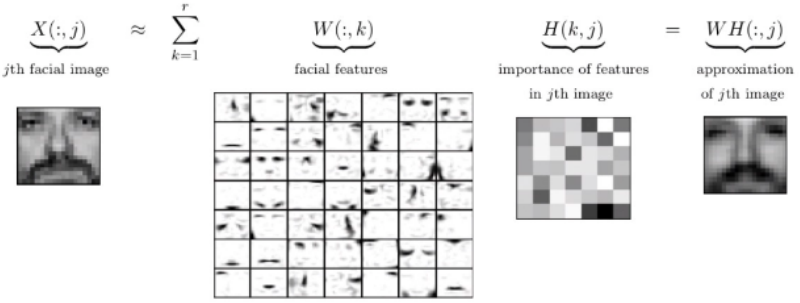

Figure 1: one real example of W and H

1.1 Main idea of NMF

In real life, data is often non-negative by nature, such as image intensity, movie ratings, document-tern counts[2]. The main idea of NMF is to factorise a non-negative high-dimension matrix into the product of two non-negative low-dimension matrices. One matrix is a dictionary matrix, it can be interpreted as images (the basis images). The other is the activation matrix, it tells us how to sum up the basis images in order to reconstruct. Figure 2 gives us an example to truly understand the matrix of these two substitute matrices[3].

1.2 problem intended to solve and why it is important

This assignment has 2 datasets, ORL dataset and Extended YaleB dataset. ORL dataset contains 400 images of 40 distinct subjects. Extended YaleB dataset contains 2414 images of 38 subjects under 9 poses and 64 illumination conditions. For each dataset, two types of noise would be added to stimulate the noised images that this experiment is intended to test the robustness of the NMF algorithm on noisy image data.

Compared with other two traditional decomposition methods VQ and PCA, the dictionary matrix in the VQ reflects the combination of the basic features in each image, the activation matrix is sparse and the reconstruction image is one-to-one correspondence to the dictionary. In this case, NMF usually has better performance. For PCA, the activation matrix could be negative, so PCA has no intuitive physical meaning. NMF has a very good interpretability and it has very promising prospects.

The robustness of the NMF algorithm means how tolerant the algorithm is to different types of noise. After comparing the performance of different algorithms in different types of noise, the robustness of each algorithm can be obtained. We can know which algorithm is suitable for solving most of the noise. When meeting an unknown noise dataset, this algorithm can be considered as the first choice to recover this dataset. After analysing the robustness, we can obtain the knowledge of how good each NMF algorithm is at recovering noise.

1.3 NMF applications

One of the most successful applications in NMF is image analysis and processing. The images often include a mass of data and the computer usually saves the image information based on the matrix. Image recognition, analysis and processing are also based on the matrix. These features let NMF algorithms suit the image analysis and processing. For example, NMF algorithm has been used to automatically recognize the garbage debris in space by processing the image sent back from the satellite. Another example, the NMF algorithm has been used to identify stars by processing the images taken by astronomical telescopes. Also, the US uses identification systems related to the NMF algorithm at airports to automatically identify terrorists entering and leaving the airport based on the image collections of the terrorist profiles. Other than these, NMF algorithm also has a wide variety of applications in the field of data mining, speech processing, robot control, biomedical engineering and chemical engineering.

1.4 Overview of the NMF methods

L_2 NMF can make the decomposition matrix all non-negative and achieve dimension reduction. All variants of the NMF algorithm are derived from the L_2 NMF algorithm.

The L_2 norm NMF is easy to be impacted by the outliers and noises.To increase the performance of NMF algorithm, $L_{2,1}$ norm based regularisation has been established as a novel, efficient and robust means to apply to domains sensitive to outliers [4].

Since the L_2 and $L_{2,1}$ Norm NMF algorithms do not have a good performance in some complex datasets, CIM-NMF algorithm is implemented. This algorithm adds a more complex term which is good at handling the datasets with more outliers and noises.

2 Previous work

NMF algorithm is a matrix decomposition method proposed by Lee and Seung in 1999 in the journal Nature[5]. Research has shown that part-based representations exist in the brain or computer, and the authors ask the question, is the perception of the whole based on the perception of its parts. The authors demonstrated an algorithm for non-negative matrix factorization which can derive part-based representations. Two matrices were derived from the original matrix, one matrix represents the part-based representation, the other matrix represents the weight of part-representation in each image. The authors found that the NMF algorithm can generate an observable representation based on each part-based representation. It also can achieve dimension reduction. NMF has a broad potential application in many fields. They demonstrate the basic expression. All subsequent variants of the NMF algorithm are derived from this basic NMF algorithm.

The basic definition of the NMF will be shown below: Given a non-negative matrix $X \in R_{mxn}$ the purpose of NMF is to find two substituted non-negative matrix $U \in R_{mxk}$ and $V \in R_{kxn}$, let $X \approx UV$, and the objective function of NMF is

$$\min_{U \in u, V \in v} ||X - UV||_F^2$$

where $||.||_F$ indicates the Frobenius norm. The iterative multiplicative update rule (MUR) below is used to minimise the objective function

$$V \leftarrow V \times \frac{U^T X}{U^T U V}$$
$$U \leftarrow U \times \frac{U^T X}{U V V^T}$$

3 Methodology

3.1 Noise Algorithm

3.1.1 Block-occlusion

The method for adding block-occlusion noise to the image is to corrupt a fixed shape of image pixels values. In this experiments, the noise is set in a shape of square and the image pixel values to the brightest as 255. The noise is generated by randomly setting a pixel as the top left corner of the square and turn all the pixel value in the related square into the 255. The parameter to tune the block-occlusion noise is the value "b" as it change the edges' length of the square. For example a block with parameter "b" of 7 set a square of pixel value 255 with the edges' length of 7 pixel. The Figure.4 show the block occlusion noise applied on one of the image from ORL dataset.

3.1.2 Salt and Pepper

The method for adding salt and pepper noise to the image is to randomly corrupt a pixel from the image space and set the value of the pixel to either 255 or 0 and result an image looks like it got spread on with salt and pepper. The parameter to tune this method is "p". Tune the "p" will result in change of proportion of the image pixels that is corrupted and in this experiment we fixed the proportion of the noise been set to 0 as 50%. For example a salt and pepper noise with "p" set to 0.1 will result an image that have 10% of its image pixels been corrupted. The corrupted pixel have a 50% chance of being white pixel and 50% chance of being black pixel. The Figure.5 show the salt and pepper noise applied on one of the Image from ORL dataset.

3.2 NMF Algorithm

3.2.1 L_2 Norm NMF

As the introduction above, the main idea of NMF is to factorise a non-negative high-dimension matrix into the product of two non-negative low-dimension matrices. Given a non-negative matrix $X \in R_{mxn}$, the colunm of X represents a sample(in this case, each column represents a face), the row of the X represents the pixel with the same place in each sample, the purpose of L_2 NMF is to

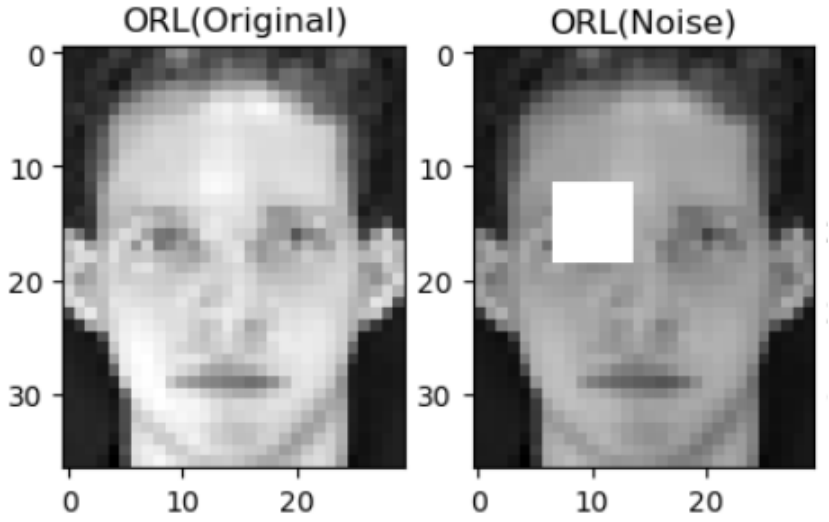

Figure 2: Block-occlusion noise example

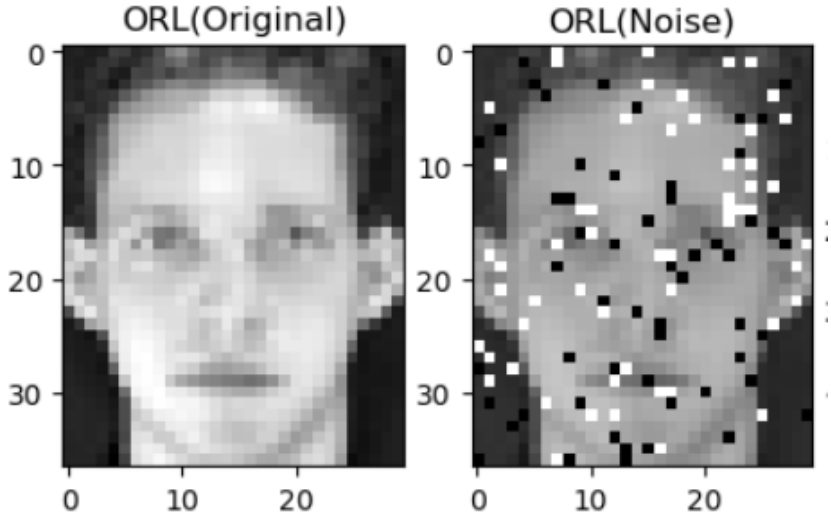

Figure 3: Salt and Pepper noise example

find two non-negative matrix $U \in R_{mxk}$ and $V \in R_{kxn}$, the product of matrix U and V is used to infinitely approach to X, so we can get the function below:

$$X \approx UV$$

Matrix U represents a dictionary matrix which includes the part-based representation, matrix V tells us the weight of the part-based representation in each image. The product of the matrix U and V represents the reconstruction image. The error function and Frobenius Norm is defined as below:

$$min||X - UV||^2, w.t, U, V; s.t. U, V > 0$$

$$||X||_F = \sqrt{trace(X^T X)} = \sqrt{\sum_{i=1}^{d} \sum_{j=1}^{n} X_{i,j}^2}$$

The objective function is defined as:

$$\min_{U \in u, V \in v} ||X - UV||_F^2$$

It is convex in matrix U or V but not in both. An algorithm called multiplicative update rule(MUR) was introduced by Lee and Seung in 1999[5]. The main idea for this algorithm is to fix matrix U, then solve matrix V. After suitable iteration, then a local minimum can be found. The original gradient descent function is as follow:

$$V \leftarrow V - \eta \times [U^T UV - U^T X]$$
$$U \leftarrow U - \zeta \times [U^T UV - U^T X]$$

Lee and Seung have proved that under these object rules, gradient descent ensures the convergence of this problem in a limited number of iterations[6]. They ingeniously set the learning rate that can easily ensure that all values are non-negative:

$$\eta \rightarrow \frac{V}{U^T UV}$$
$$\zeta \rightarrow \frac{U}{UVV^T}$$

After taking this learning rate to the gradient descent function, finally, the multiplicative update rules can be shown as:

$$V \leftarrow V \times \frac{U^T X}{U^T UV}$$
$$U \leftarrow U \times \frac{U^T X}{UVV^T}$$

3.2.2 L_{21} Norm NMF

The L_2 norm NMF is easy to be impacted by the outliers and noises, so in the practical applications, it may contribute to a bad performance. By applying the $L_{2,1}$ NMF algorithm, this improved the NMF algorithm's ability to deal with outliers or noises. The definition of $L_{2,1}$ norm has been given:

$$||X||_{2,1} = \sum_{i=1}^{n} \sqrt{\sum_{j=1}^{p} X_{i,j}^2} = \sum_{i=1}^{n} ||X_i||$$

Based on the definition of L_{21} norm, the error function of $L_{2,1}$ NMF can be derived as follow:

$$||X - UV||_{2,1} = \sum_{i=1}^{n} \sqrt{\sum_{j=1}^{p} (X - UV)_{j,i}^2} = \sum_{i=1}^{n} ||x_i - Uv_i||$$

The objective function can be derived as follow:

$$\min_{U,V} ||X - UV||_{2,1} \; s.t. U \geq 0, V \geq 0$$

In order to find the minimum of the objective function, Kong (2011) extends the L_2 MUR optimisation method, with the addition of a weighted matrix regulariser D [7]. The regulariser D is used to decrease the influence of the outliers. The regulariser D and the update function are shown as follow:

$$D_{ii} \Leftarrow \frac{1}{\sqrt{\sum_{j=1}^{p} (X - UV)_{ji}^2}} = \frac{1}{||X_i - Uv_i||}$$
$$U_{jk} \Leftarrow U_{jk} \frac{(XDV^T)_{jk}}{(UVDV_T)_{jk}}$$
$$V_{ki} \Leftarrow V_{ki} \frac{(U^T XD)_{ki}}{(U^T UVD_T)_{ki}}$$

This paper[7] also gives the deduction of the $L_{2,1}$ updating method. First, the diagonal matrix D_{ii} is calculated. The elements of D_{ii} provide the weight to regularise the outliers in the dataset. The initial values of U_{jk} and V_{ki} are assigned randomly. For the iteration, the value of V is fixed, then update the value of U. After that, the value of U is fixed, then update the value of V. Stop the iteration when the tolerance is met.

3.2.3 CIM-NMF

According to paper [8], the Half-Quadratic Minimization is used as a cost function for CIM-NMF. The formula of the Half-Quadratic based augmented objective function is given as:

$$\min_{U,V,W} \sum_{i=1}^{N} \sum_{j=1}^{M} [W_{ij}(X_{ij} - \sum_{k=1}^{K} U_{ik}V_{kj})^2 + \phi(W_{ij})]$$

In this augmented objective function W_{ij} is the corresponded auxiliary variable, where as the $\phi(W_{ij})$ is the conjugate function of the $l(E_{ij})$[8]. Since the $l(E_{ij})$ itself is not convex. The conjugate function has the similar property with the original function, but it is convex. Here the conjugate function is chosen to replace the true loss. For each update, when the matrices U and V are fixed, the optimal value of W_{ij} is given by:

$$W_{ij} = \frac{\frac{d}{dE_{ij}}(1-g(E_{ij},\sigma))}{E_{ij}} \propto exp(-\frac{(X_{ij}-\sum_{k=1}^{K} U_{ik}V_{kj})^2}{2\sigma^2})$$

The optimization process can be explained as follows, for a data matrix $X \in R$ with N rows and M columns, we firstly initialise two matrices U (N rows and K columns) and V (M rows and K columns), with random values. Then, for the defined U and V, we first calculate kernel size of the data, take the data matrix X as the average, subtract the reconstructed value of the corresponding pixel, square it, sum the result of all pixels, and then divide by the two times total number of pixels. After calculating kernel size, we can calculate the update weights W_{ij} of different pixels through the kernel size. By updating the weights, we can perform non-negative matrix updates of U and V in the previous step. For the updated U and V, we will judge the difference between the reconstructed matrix and the original matrix X again. When the difference is small or the maximum number of iterations is reached, the non-negative matrix factorization algorithm will be finished. Otherwise, we need to calculate the kernel size and start a new round of updates again.

The algorithm process is shown below:
Calculate the kernel size of the reconstructed matrix:

$$\sigma^2 = \frac{1}{2NM} \sum_{i=1}^{N} \sum_{j=1}^{M} (X_{ij} - \sum_{k=1}^{K} U_{ik}V_{kj})^2$$

Calculate the weight matrix W through the U, V and the kernel size:

$$W_{ij} = exp(-\frac{(X_{ij}-\sum_{k=1}^{K} U_{ik}V_{kj})^2}{2\sigma^2})$$

Fix V and update U by

$$U_{ik} = U_{ik} \frac{(W \otimes XV)_{ik}}{(W \otimes (UV^T)V)_{ik}}$$

Fix U and update V by:

$$V_{kj} = V_{kj} \frac{((W \otimes X)^T U)_{kj}}{((W \otimes (UV^T))^T)_{kj}}$$

Back to Step 1 until Converges.

In conclusion, after inputting the initial values of matrices U and V, here the matrices U and V are fixed, then we calculate the matrix W. After that the new matrices U and V can be updated. Finally, the σ^2 will be updated in order to the next iteration. Repeat these steps again and again until the tolerance is met.

3.2.4 Discussion Among Three different techniques

L_2 NMF
This is the most basic and the first proposed NMF algorithm. It has the lowest complexity and other NMF algorithms are derived by L2 NMF. This algorithm uses the Frobenius norm in the objective function. The calculation on the squared loss is easy to be influenced by the outliers and noises.

L_{21} NMF
Since L2 NMF is susceptible to the noise and outliers, people implement the L21 NMF. This algorithm adds a weighted matrix regulariser in the update function, although it increases the complexity of the algorithm, it also changes the square error to the absolute error, which makes the algorithm better handle data with more noise.

CIM-NMF
Compared with L2 NMF, the CIM NMF adds an auxiliary variable W and a conjugate function of the loss function. Although it has the highest complexity of these 3 algorithms, it also ensures that the algorithm is the most robust.

4 Experiments and Discussions

4.1 Datasets

The datasets are used to perform these experiments are ORL dataset and YaleB dataset.
The ORL dataset contains 400 images of 40 distinct subjects and in different angle and lighting with different facial expressions and details. To reduce the computational complexity, all the images are resized to 30*37 pixels.
The YaleB dataset contains 2414 images of 38 subjects in different poses and under different lighting conditions. The original image size was 168 x 192 pixels; all images are resized to 42 x 48 pixels to reduce computational complexity.

4.2 Noise setup

Two different noises are created on the dataset. One is block noise and the other is salt and pepper noise. For block noise, we set up a value b, which refers to the edge length of a square that will be set as a white block. That is, we randomly generate a white square area with a set edge length of b for each image. When setting up the experiments, we build two different types of block noise on the dataset, with b = 7 and b = 14 respectively to simulate the low noise and high noise situation. The other noise is salt and pepper noise. This kind of noise is generated by randomly selecting pixels and setting them as white or black with a set percentage. Similar to block noise, we use p value to control the ratio of the noise pixel. The larger the p value is, the more pixels will be turned to white or black. When setting up experiments, we use two p values, 0.1 for low noise level and 0.4 for high noise level.

In the first set up, we test our NMF algorithms with a small noise, that is b = 7 and p = 0.1, and the second set up, we test our NMF algorithm with a large noise, that is b = 14 and p = 0.4. The reason for this noise set up is to test if the robustness holds true when the algorithm encounters different strength of noises.

4.3 Experiments design

In order to comprehensively test the performance of the NMF algorithm, we set two groups of noise points with different degrees, and add corresponding noise points to the two data sets.

In each set up, L_2 Norm NMF, L_{21} Norm NMF and CIM-NMF are used to decompose the data matrix with noise. In order to better reveal the performance of the model, we use three indicators, namely RRE, ACC and NMI. RRE refers to relative reconstruction errors. It calculates the proportion of the difference between the reconstructed matrix and the original matrix to represent the performance of the NMF model. Smaller the RRE, better the performance. ACC refers to accuracy, which is the proportion of reconstructed images whose labels are predicted correctly using k-mean cluster to get the prediction then calculate the average accuracy. The last NMI refers to Normalized Mutual Information, which reflects the performance of NMF by calculating the entropy change between the true value and the predicted value.

Each algorithm is updated with a maximum of 300 iteration, where the algorithm can stop when it hits the tolerance threshold. In order to get a rigorous performance evaluation. Each algorithm is tested 5 times on the same dataset. In the second experiment, we use the same procedures, except that the datasets are replaced by the datasets with large noise added.

Table 1: Reconstruction error of NMFs

Experiment	ORL		YaleB	
Block Noise b = 7	Mean	S.d	Mean	S.d
L_2-norm NMF	0.235	0.002	0.321	0.003
$L_{2,1}$-norm NMF	0.239	0.002	0.319	0.002
CIM-NMF	0.145	0.001	0.207	0.002
Salt and Pepper noise p=0.1	Mean	S.d	Mean	S.d
L_2-norm NMF	0.265	0.004	0.258	0.004
$L_{2,1}$-norm NMF	0.271	0.002	0.267	0.004
CIM-NMF	0.122	0.001	0.183	0.002
Block Noise b = 14	Mean	S.d	Mean	S.d
L_2-norm NMF	0.448	0.005	0.628	0.010
$L_{2,1}$-norm NMF	0.446	0.005	0.630	0.006
CIM-NMF	0.318	0.004	0.237	0.005
Salt and Pepper noise p=0.4	Mean	S.d	Mean	S.d
L_2-norm NMF	0.554	0.003	0.578	0.004
$L_{2,1}$-norm NMF	0.534	0.008	0.579	0.005
CIM-NMF	0.394	0.009	0.259	0.002

Table 2: accuracy score of NMFs

Experiment	ORL		YaleB	
Block Noise b = 7	Mean	S.d	Mean	S.d
L_2-norm NMF	0.514	0.020	0.237	0.0175
$L_{2,1}$-norm NMF	0.514	0.042	0.247	0.016
CIM-NMF	0.720	0.015	0.275	0.011
Salt and Pepper noise p=0.1	Mean	S.d	Mean	S.d
L_2-norm NMF	0.396	0.019	0.188	0.013
$L_{2,1}$-norm NMF	0.399	0.020	0.159	0.008
CIM-NMF	0.730	0.021	0.292	0.020
Block Noise b = 14	Mean	S.d	Mean	S.d
L_2-norm NMF	0.336	0.028	0.013	0.003
$L_{2,1}$-norm NMF	0.333	0.012	0.130	0.008
CIM-NMF	0.525	0.043	0.278	0.020
Salt and Pepper noise p=0.4	Mean	S.d	Mean	S.d
L_2-norm NMF	0.333	0.015	0.108	0.003
$L_{2,1}$-norm NMF	0.292	0.009	0.108	0.002
CIM-NMF	0.402	0.011	0.193	0.005

Through these two control experiments, we can not only compare the performance of different NMF horizontally, but also compare the robustness of the same algorithm to different levels of noise longitudinally.

4.4 Results of our experiments

The result tables for relative reconstruction error is table 1, accuracy score is table 2 and normalised mutual information is table 3.

Our experiments' results are calculated through each set up and get the average of the resulted RRE, ACC and NMI, these three metrics with there respective standard deviation. By observing all three tables' of metric, there is no significant standard deviation for all results. Therefore, we can relatively safely assume the rigorous performance evaluation had achieved.

Table 3: Normalized Mutual Information of NMFs

Experiment	ORL		YaleB	
Block Noise b = 7	Mean	S.d	Mean	S.d
L_2-norm NMF	0.665	0.022	0.311	0.014
$L_{2,1}$-norm NMF	0.662	0.042	0.306	0.018
CIM-NMF	0.845	0.009	0.326	0.010
Salt and Pepper noise p=0.1	Mean	S.d	Mean	S.d
L_2-norm NMF	0.563	0.016	0.259	0.011
$L_{2,1}$-norm NMF	0.587	0.016	0.212	0.017
CIM-NMF	0.848	0.019	0.356	0.021
Block Noise b = 14	Mean	S.d	Mean	S.d
L_2-norm NMF	0.510	0.027	0.146	0.012
$L_{2,1}$-norm NMF	0.501	0.019	0.154	0.014
CIM-NMF	0.678	0.034	0.330	0.021
Salt and Pepper noise p=0.4	Mean	S.d	Mean	S.d
L_2-norm NMF	0.437	0.017	0.102	0.004
$L_{2,1}$-norm NMF	0.388	0.010	0.102	0.005
CIM-NMF	0.589	0.013	0.255	0.004

In our first light noise scenario, L_2 norm NMF and $L_{2,1}$ norm NMF performed relatively the same as they have similar measurements across all three metrics for each dataset and we can clearly see a major jump on performance when it came to CIM-NMF as it consistently have lower Reconstruction error on every single setting, higher accuracy and higher NMI. Another observation is that both L_2 and $L_{2,1}$ have slightly better performance on dealing with the block noise, while CIM-NMF have better performance when dealing with salt and pepper noise.

On the heavy noise scenario, we observe the same trend in terms of general performance as the light noise scenario, all models have suffered a performance loss at the same ratio, as the CIM-NMF still holds the best performance amount the three. As this observation justify that under both light and heavy noise scenario, for each type of noise we test, the CIM-NMF model consistently perform the best.

These results are not exactly consistence with our expectation based on the methodology as the $L_{2,1}$ Norm NMF was expected to perform better than L_2 Norm NMF but in experiment they have similar performance.

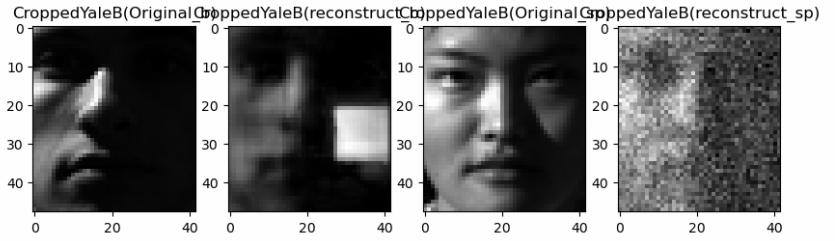

Figure 4: L_2 Norm NMF reconstruction image compare to original image.

By comparing the different examples of reconstruction image of figure 4,5 and 6, it is quite easy to tell that the CIM-NMF have a better reconstructed image even under the heavy noise situation, as the face are more visible than other two NMFs.

The block noise is almost not visible on the CIM-NMF reconstructed image while other two have a clear outline of the while square.

The salt and pepper noise is drastically less obvious for the CIM-NMF reconstructed image and has produced great facial features that is similar to original image while other two NMFs fail to do so.

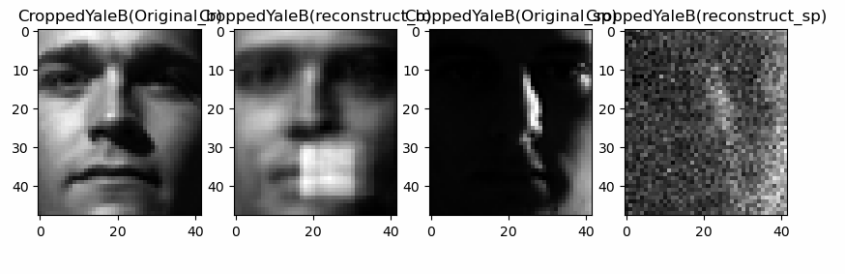

Figure 5: $L_{2,1}$ Norm NMF reconstruction image compare to original image.

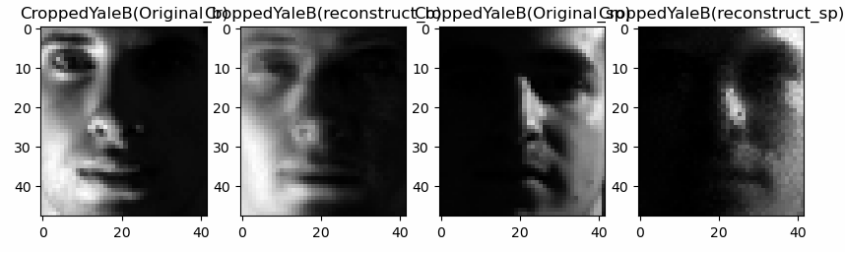

Figure 6: CIM-NMF reconstruction image compare to original image.

5 Conclusions and Future work

In conclusion of our experiment, judging the result output of all three metrics, three NMFs performed almost twice as good when encounter low value/proportion of noise than high value/proportion of noise as expected. However the performace of $L_{2,1}$ Norm NMF is below our expectation since it perform similarly to L_2 metrics wise but also visually the L_2 provide a even better image with better facial recognition.

Finally, it is quite easy to tell that CIM-NMF is the NMF which more robust towards the noise as it show good performance under both light noise and heavy noise situation no matter which type of noise been applied.

Also by observing the reconstructed image of CIM-NMF, it does show a visually closer representation of the original image than other two NMFs do, which holds true to our metrics' results.

This experiment's have few limitations, for both datasets the images were resized to a lower pixel values to reduce the computational cost. As this operation would result in lowering the image quality of the dataset, thus it would negatively impact each NMF's performance differently, as this might affect the correctness of the experiment when the robustness of model was judged purely based on the metrics. Another limitation is the experiment environment, as the model used to compute images generally has a high computational cost and for our models we have to sacrificed some model performance in order to keep the run-time in a reasonable margin.

5.1 future work

In future, we can extend on the current experiments in following aspects:
1. Investigate the negative influences of the low resolution images on each model.
2. Using better experiment environment such as gpus, thus models can have a narrower tolerance level and higher maximum iteration limits, therefore the full potential of the model can be achieved.
3. Using different initialisation methods other than random initialisation, such as NNDSVD initialisation. 4. Extend the noise designs, such as different noise size and different noise type like Gaussian noise.

References

[1] Fei Duan, Hui-Min Wang, Chao Zhang. (2021). Cauchy Non-negative Matrix Factorization for Data Representation. Ji suan ji ke xue, 48(6), 96–.

[2] Liu, T., 2022. Advanced Machine Learning (COMP 5328) Dictionary Learning and Non-negative Matrix Factorisation.

[3] Colyer, A. (2022). The why and how of nonnegative matrix factorization. Retrieved 2 October 2022, from https://blog.acolyer.org/2019/02/18/the-why-and-how-of-nonnegative-matrix-factorization/: :text=NMF

[4] Feiping Nie et al. "Efficient and robust feature selection via joint l2,1-norms minimization". In: In NIPS. 2010.

[5] Seung, H. S., Lee, D. D. (1999). Learning the parts of objects by non-negative matrix factorization. Nature (London), 401(6755), 788–791. https://doi.org/10.1038/44565

[6] Alex Díaz, Damian Steele. (2021). Analysis of the robustness of NMF algorithms. arXiv.org.

[7] 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. Glasgow, Scotland, UK: Association for Computing Machinery, 2011, pp. 673–682. DOI: 10.1145/2063576.2063676. URL: https://doi.org/ 10.1145/2063576.2063676.

[8] Liang Du, Xuan Li, Yi-Dong Shen. (2012). Robust Nonnegative Matrix Factorization via Half-Quadratic Minimization. 2012 IEEE 12th International Conference on Data Mining, 201–210. https://doi.org/10.1109/ICDM.2012.39

6 Appendix

6.1 Code file instruction

This submission will include 2 code file in the form of .ipynb and an empty data folder.
The 2 code file is in the name of "Main_code" as all the algorithm will be run through this file and the "Out_put_calculation" for calculating the mean and standard deviation of the 5 experiment for each algorithm.
In order to run the file"Main_code" please put the data folder and the code file in the same level of the folder structure and please remove the section of "zip file unzip" with the first code chunk that deal with jupyter note book upload data folder issue and continue with the rest of the code.
The code file "Out_put_calculation" do not need any pre-requirements or adjustments, you can run it from top to bottom to get the result of the experiments.

6.2 Individual Contribution

Qi Lu: NMF algorithm implementation of L_2 and $L_{2,1}$, the experiment design and result output implementation.
Report Methodology noise section, experiments and discussions datasets, results and experiments,and conclusion and Future work sections.
Report latex editing and proof reading.

Shiming Guo: program two types of noise function and draw the schematic of original and noise image in the code.
Write the abstract, introduction, previous work, the whole details of 2 NMF algorithm, the part details of 1 NMF algorithm and discussion among 3 NMF algorithm

Zheng Wang: In the code section, firstly, the programming of all the code in the CIM-NMF section was completed. Secondly, the finalisation of the code section was completed and the structure of the entire code file was finalised. In the paper section. Firstly, the CIM-NMF part of the NMF

Algorithm was completed. Second, completed the 'Datasets' and 'Experiments design' sections of Experiments and Discussions