COMP 5328

Assignment1 Report

Group17

Group Members

SID	Name
500230978	Tiannan Chen
490169971	Haochen Zhang

Abstract:

Nonnegative matrix factorization (NMF) is a widely used technique in face recognition. It decomposes a nonnegative large matrix into two nonnegative small matrices, and it uses the product of two nonnegative matrices to approximate the original nonnegative data in the high-dimensional space by a linear representation in the low-dimensional space. Moreover, the data is usually corrupted by the accompanying noise. This assignment aims to implement three different NMF algorithms in Python, apply them to block-occlusion noise grayscale face images, and compare their robustness. Finally, it was run in the Jupyter Notebook environment, and only sklearn was used to evaluate each algorithm.

Introduction:

In this experiment, we want to solve the problem of implementing three different Non-negative Matrix Factorization algorithms without using external packages such as scikit-learn and Nimfa. The robustness of grayscale image data under the influence of block-occlusion noise is also analyzed. It is difficult to distinguish the pictures affected by noise, so how to deal with the noise of the image this problem is particularly important, and NMF can be a good feature extraction in noisy images, and is widely used in image analysis and processing. The experiment implements three different Non-negative Matrix Factorization algorithms, namely Simple NMF, L1 norm based NMF and L2,1-norm based NMF.

Related work:

- **Simple Non-negative matrix factorization (NMF)**
  The main idea in the simple nonnegative matrix factorization algorithm is to update the algorithm with multiplication, which aims to factorize a given matrix into two nonnegative matrices, P and Q, using an iterative multiplicative update rule. Its goal is to approximate the original matrix R as closely as possible.
  The advantages of simple NMF are that it is easy to implement among the three algorithms and takes the least time. The disadvantages are noise sensitivity, and high dimensions of the feature may lead to decreased accuracy.

- **L1 norm based NMF**
  In L1 norm-based NMF, the main idea is to add the product of the L1 norm and regularization strength to the original simple NMF algorithm in each iteration to produce a more sparse matrix. L1 norm-based NMF has the advantage of helping

feature selection but has the disadvantage of affecting the selection of regularization strength. Moreover, the processing time is longer than normal NMF.

- **L21 norm based NMF**

  In L2,1 norm-based NMF, the main idea is to add the product of L2,1 norm strengths to the original simple NMF algorithm in each iteration to produce a sparser matrix . NMF based on the L2,1 norm helps in row feature selection, besides, the processing time is longer than standard NMF and L1 NMF. Standard NMF is unstable with respect to noisy and outliers, L21 NMF can adapt to outliers and noise better than standard methods(Deguang Kong,2011).

Methods:

- **Choices of noise**

  In this experiment, the noise selected is block-occlusion noise, which is a kind of noise that partially blocks the image, and the occluded part is a square, thus interfering with the integrity of the image. It is more meaningful and practical when comparing the robustness of the three NMF algorithms. In the experiment, the side length b of block-occlusion noise block is set to 10 and 14, and the noise block is set to a pure white square, so the pixel point of each noise block in the gray image matrix is 255.

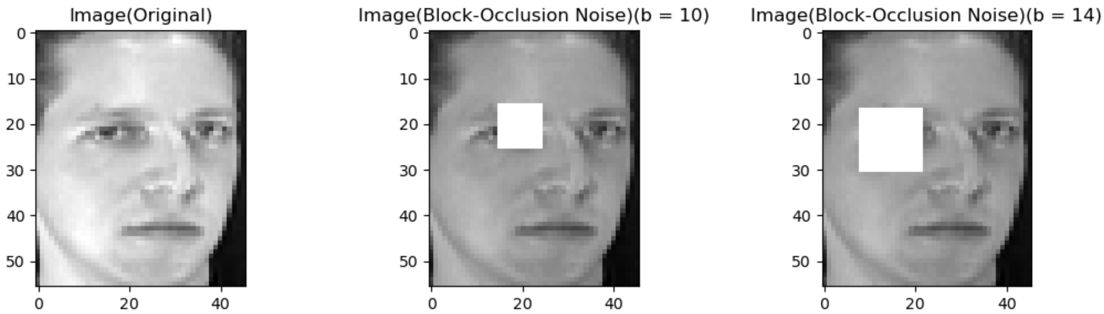


Figure 1: ORL dataset with block-occlusion noise

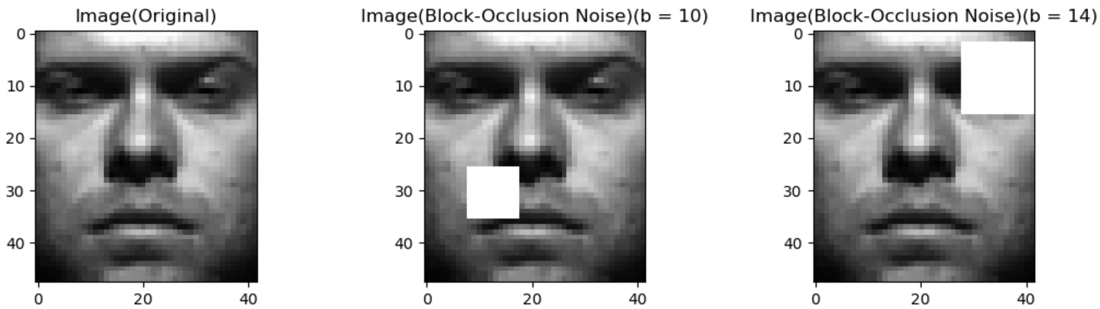
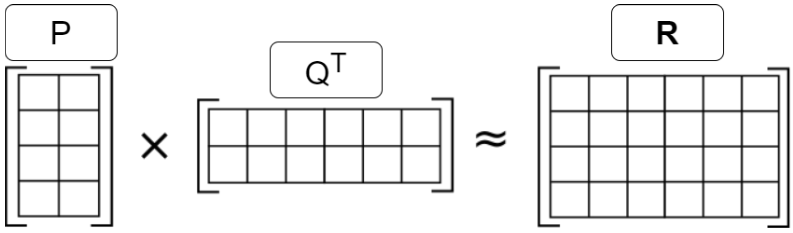


Figure 2: YaleB dataset with block-occlusion noise

- **Simple Non-negative matrix factorization (NMF)**
  In the simple Non-negative matrix factorization algorithm, given the input data is the matrix R with added block-occlusion noise, The goal is to find two nonnegative matrices P (of size m x k) and Q (of size n x k), where k is the number of latent features,



  to minimize the following loss function

$$argmin \ || \ R - PQ^T \ ||$$

  To minimize the above loss function, NMF finds the optimal P and Q by iterating the multiplicative update rule, multiplicative update rule for P,

$$\mathbf{P}_{ij} = \mathbf{P}_{ij} \frac{(\mathbf{RQ})_{ij}}{(\mathbf{PQ}^T\mathbf{Q})_{ij}}$$

  and multiplicative update rule for Q

$$\mathbf{Q}^T_{ij} = \mathbf{Q}^T_{ij} \frac{(\mathbf{P}^T\mathbf{R})_{ij}}{(\mathbf{P}^T\mathbf{P}\mathbf{Q}^T)_{ij}}$$

- **L1 norm based NMF**
  L1 norm based NMF adds the L1 norm to simple NMF to increase the sparsity of the matrix. It is implemented by adding the product of L1 norm and regularization strength α after multiplicative update rule P and Q. Because the sparsity of a vector x is 1 when there is only one nonzero value, and 0 when all elements are equal and nonzero, and the L1 norm is the sum of the absolute values of the elements in the vector (Hoyer, 2004), increasing the value of α strengthens the effect of regularization, which leads to a more sparse matrix after factorization. The multiplicative update rule for P with L1 norm,

$$P_{ij} = P_{ij} + \alpha \sum_{ij} |P_{ij}|$$

  The multiplicative update rule for Q with L1 norm

$$Q_{ij} = Q_{ij} + \alpha \sum_{ij} |Q_{ij}|$$

- **L21 norm based NMF**

  L2,1-norm NMF uses the norm l21, which is a method of combining l1 and l2. First, perform the l2 operation on each unique sequence, and then perform the l1 operation on these values.

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

  NMF based on l21 norm has different multiplicative updates, it requires diagonal signal D.

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

  The multiplication update formula is as follows

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

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

  D:diagonal matrix

  X:originial matrix

Experiment:

- **Data sets**

  In this experiment, the data sets used are ORL dataset and Extended YaleB dataset. The ORL dataset contains 40 faces in the interim. For each person, 10 photos are taken with different facial features, different facial features represent eyes open or closed, smiling or not smiling. In addition, the images are taken under different lighting conditions at different times, but the background is always black. Each image pixel size is 92 x 112 grayscale image, and each image is scaled down by a factor of 3 in the experiment, each image size is 30 x 37. Unlike the ORL dataset, the Extended YaleB dataset consists of 38 faces, each in nine poses, and is also much larger in pixel size, 168 x 192 pixels, which in our experiments has been scaled down by a factor of 4, or 42 x 48. The value range of a byte for grayscale images in both datasets is [0, 255].

- **Result**

  We evaluate each NMF using RMSE, Accuracy, and NMI. And we visualize the results of each iteration for different NMF.

  The number of iterations of all NMFs is 100, and we output the accuracy result and NMI result after 100 iterations. And the situations of different noise and different data sets were considered.

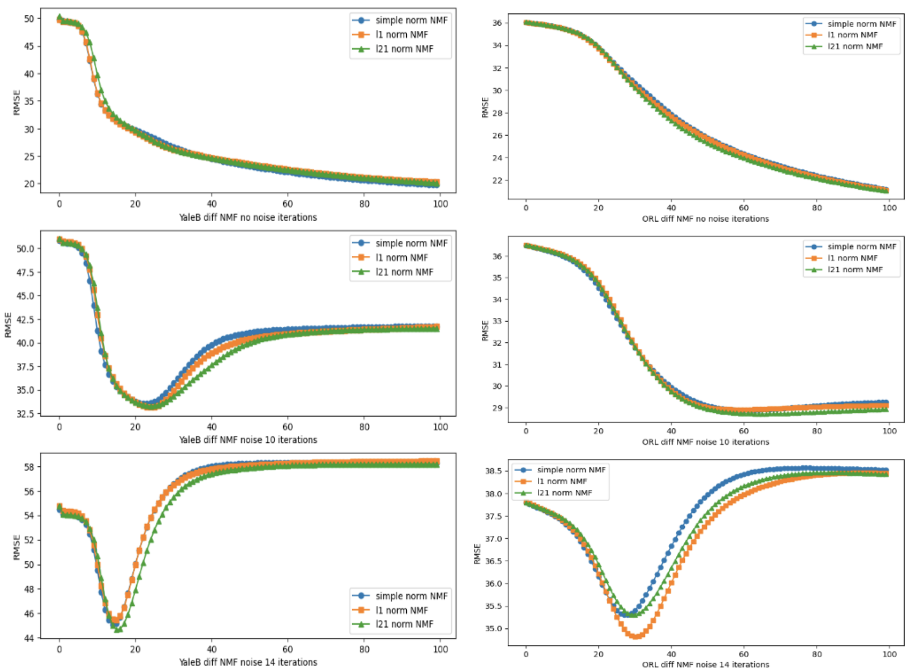
```
==> Evaluate Simple Acc and NMI ...
          --YaleB--
==> no noise
Acc(NMI) = 0.1881 (0.2695)
==>noise size 10
Acc(NMI) = 0.1015 (0.0964)
==>noise size 14
Acc(NMI) = 0.0829 (0.0747)
           --ORL--
==> no noise
Acc(NMI) = 0.7275 (0.8531)
==>noise size 10
Acc(NMI) = 0.3975 (0.5797)
==>noise size 14
Acc(NMI) = 0.2600 (0.4226)
```

```
==> Evaluate L1 Acc and NMI ...
          --YaleB--
==> no noise
Acc(NMI) = 0.1703 (0.2240)
==>noise size 10
Acc(NMI) = 0.0944 (0.0944)
==>noise size 14
Acc(NMI) = 0.0841 (0.0730)
           --ORL--
==> no noise
Acc(NMI) = 0.7025 (0.8409)
==>noise size 10
Acc(NMI) = 0.4350 (0.6178)
==>noise size 14
Acc(NMI) = 0.2375 (0.4143)
```

```
==> Evaluate L2,1-norm Acc and NMI ...
          --YaleB--
==> no noise
Acc(NMI) = 0.1686 (0.2495)
==>noise size 10
Acc(NMI) = 0.1007 (0.1148)
==>noise size 14
Acc(NMI) = 0.0899 (0.0932)
           --ORL--
==> no noise
Acc(NMI) = 0.6800 (0.8347)
==>noise size 10
Acc(NMI) = 0.4000 (0.5724)
==>noise size 14
Acc(NMI) = 0.2275 (0.3996)
```

  We can see from the above figure that in the Yale data, L21 norm NMF performs poorly when the original data or noise is relatively small, but it performs better when the noise is larger. The L2 norm NMF has better prediction results when the noise is smaller.

  We performed detailed visualizations of the data and generated a large number of images for easy comparison.Because there are too many pictures, we will display all of them in the appendix section.



By visualizing the image, we can find that L21 norm NMF exhibits lower RMSE in most cases. As the number of iterations increases, its RMSE decrease is also higher than other NMFs. And it performs very well in the presence of noise.

Conclusion:

By evaluating the data and visualizing the images we came to the following conclusions:
1. The larger the norm, the higher the prediction accuracy. The accuracy here refers to the number of accurate prediction points.
2. The larger the norm, the greater the error for inaccurately predicted points. For example, L2 needs to perform a square operation on all points, so for a point that fails to predict, its error will be expanded by the square.
3.And generally speaking, L2, 1 norm NMF is the most robust. It can adapt well to noise and produce a smaller RMSE, and its prediction effect is not worse than L2 and L1.

For future work, we have the following areas for improvement.
1. We first need more types of data sets, because the YaleB and ORL we use are relatively clear photos taken from better angles, so we need to test the processing effects of different NMFs on complex photos. And try to count the test effect on a larger data set.
2. We need to try more noise types. Different noises have different effects on each algorithm, so we need to try more noises and noises with different levels of influence.
3. Try the optimal number of iterations for each algorithm, because we found that as the iterations increase, RMSE will show a worse effect. Therefore, we plan to find a balance

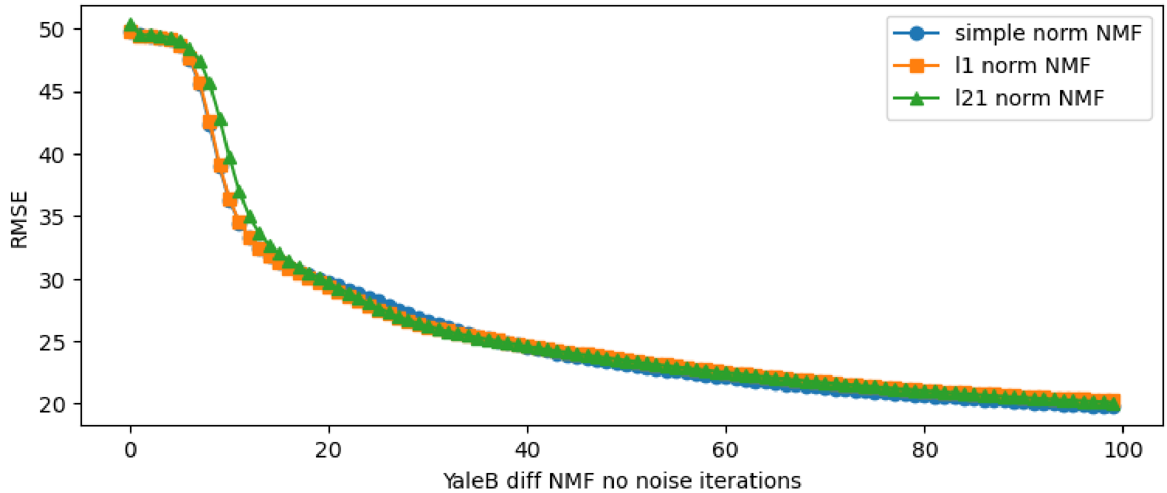
point between RMSE and accuracy, which can not only reduce the number of iterations, save costs, but also improve the effect of NMF.

Reference:

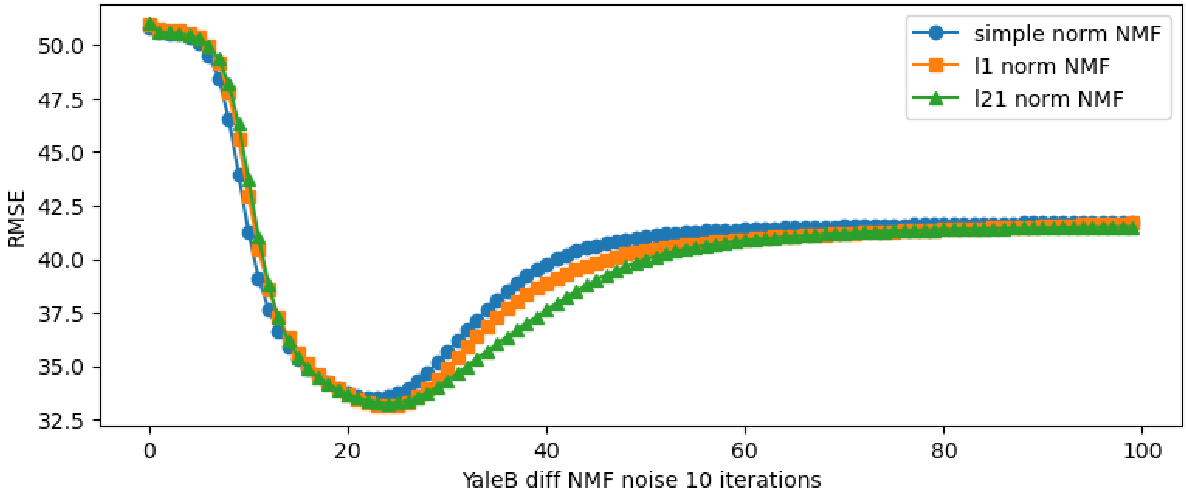
1.Hoyer, P. O. (2004). Non-negative Matrix Factorization with Sparseness Constraints. *Journal of Machine Learning Research*, *5*, 1457–1469. https://doi.org/10.5555/1005332.1044709

2.Kong, D., Ding, C., & Huang, H. (2011). Robust nonnegative matrix factorization using L21-norm. Publication History. https://do i.org/10.1145/2063576.2063676

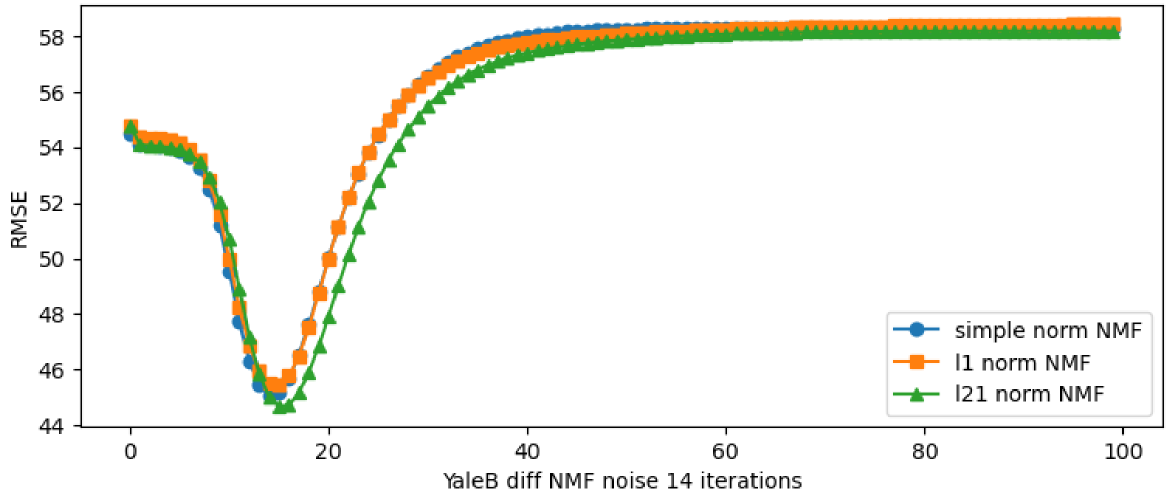
APPENDIX:



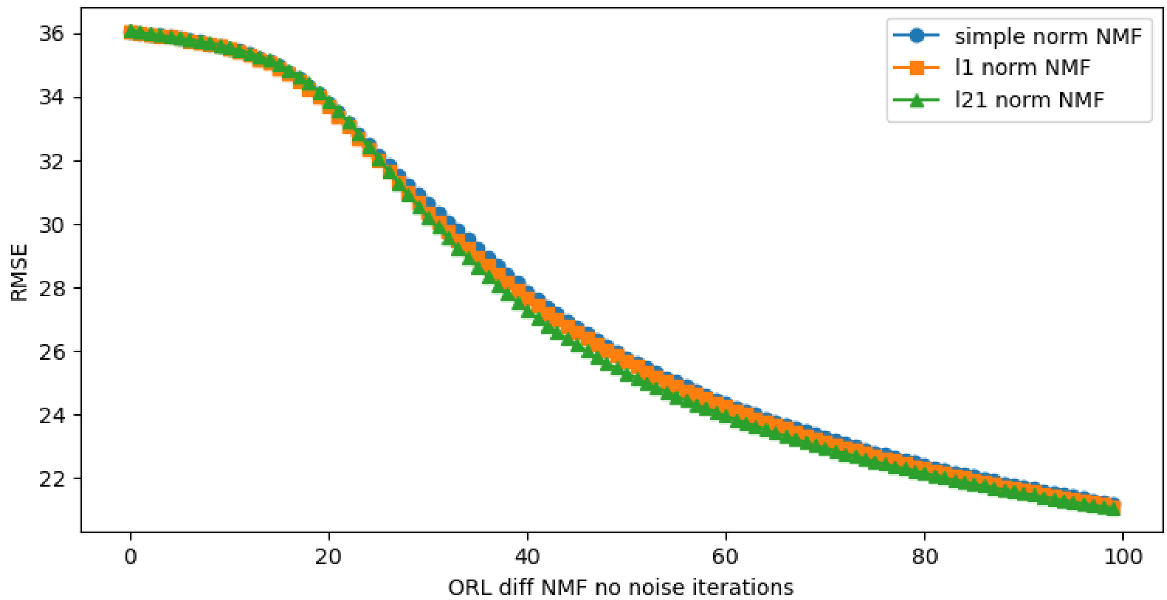
simple norm NMF
l1 norm NMF
l21 norm NMF
RMSE
50
45
40
35
30
25
20
0
20
40
60
80
100
YaleB diff NMF no noise iterations

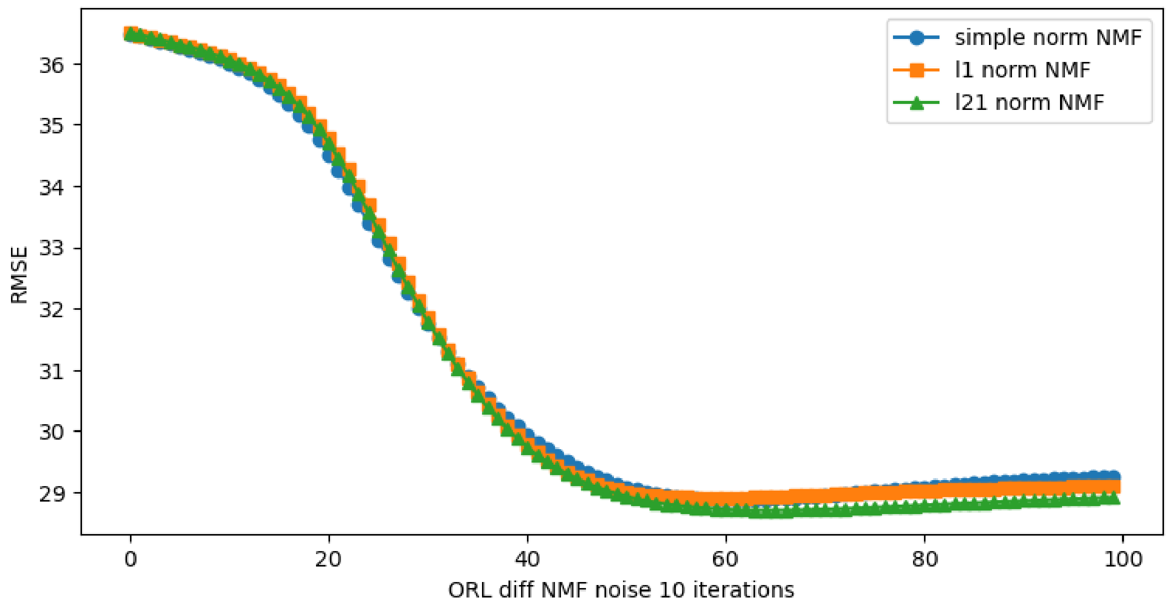


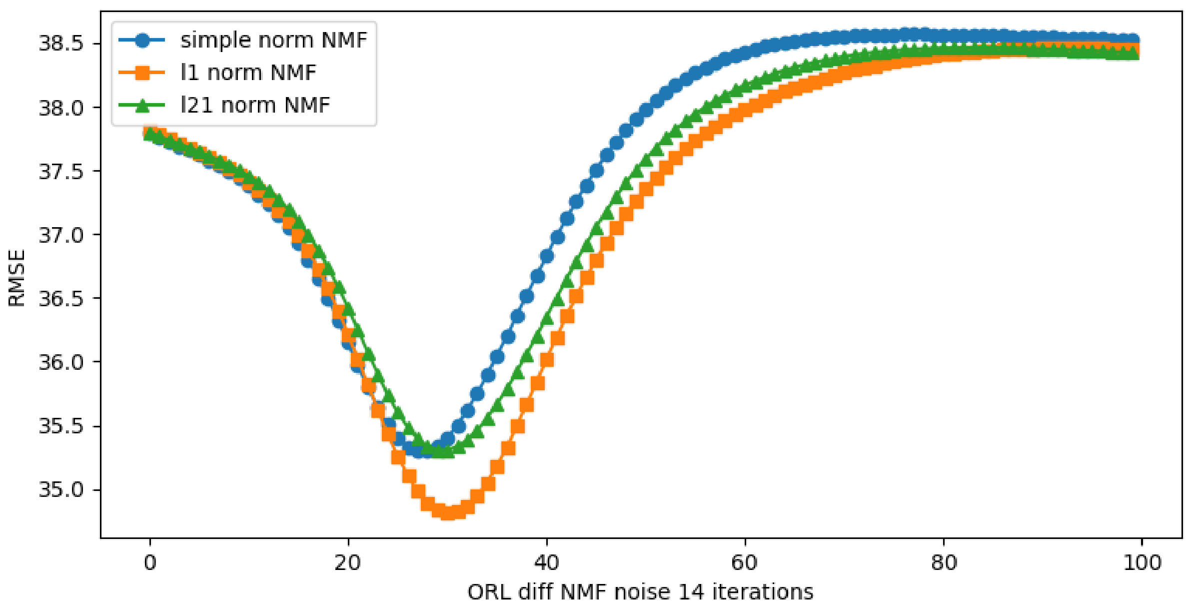
simple norm NMF
l1 norm NMF
l21 norm NMF
RMSE
50.0
47.5
45.0
42.5
40.0
37.5
35.0
32.5
0
20
40
60
80
100
YaleB diff NMF noise 10 iterations

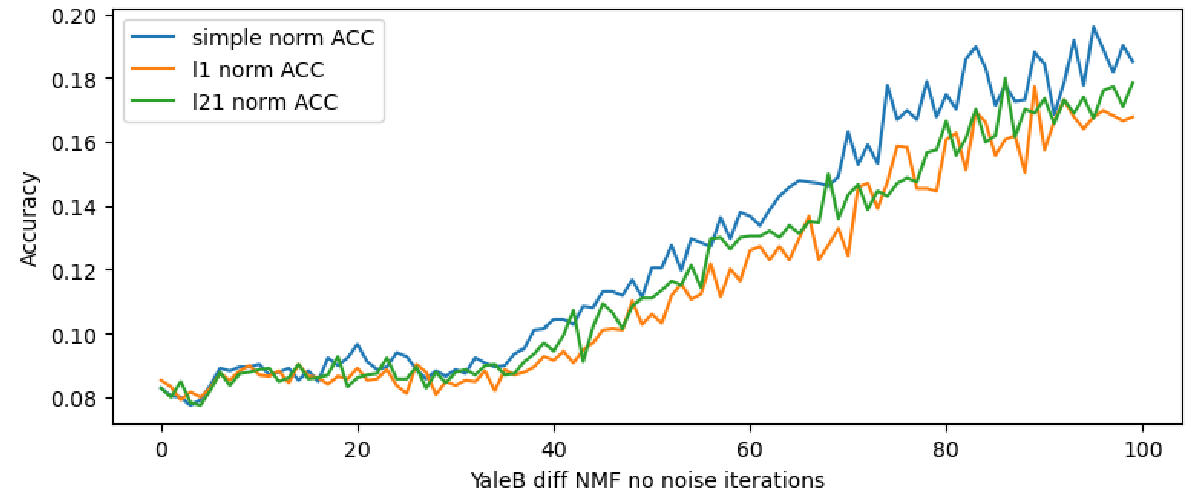


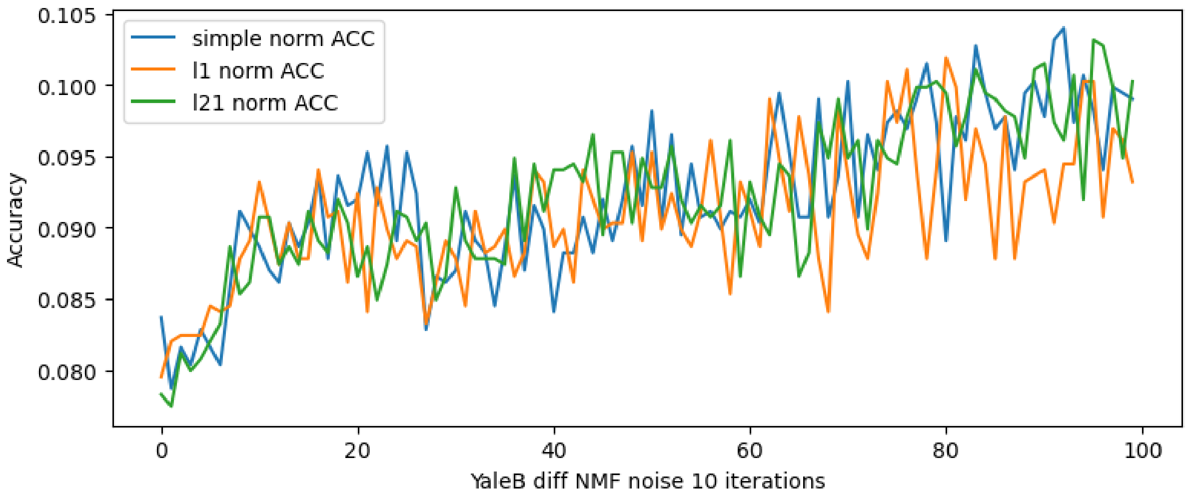
simple norm NMF
l1 norm NMF
l21 norm NMF
RMSE
58
56
54
52
50
48
46
44
0
20
40
60
80
100
YaleB diff NMF noise 14 iterations

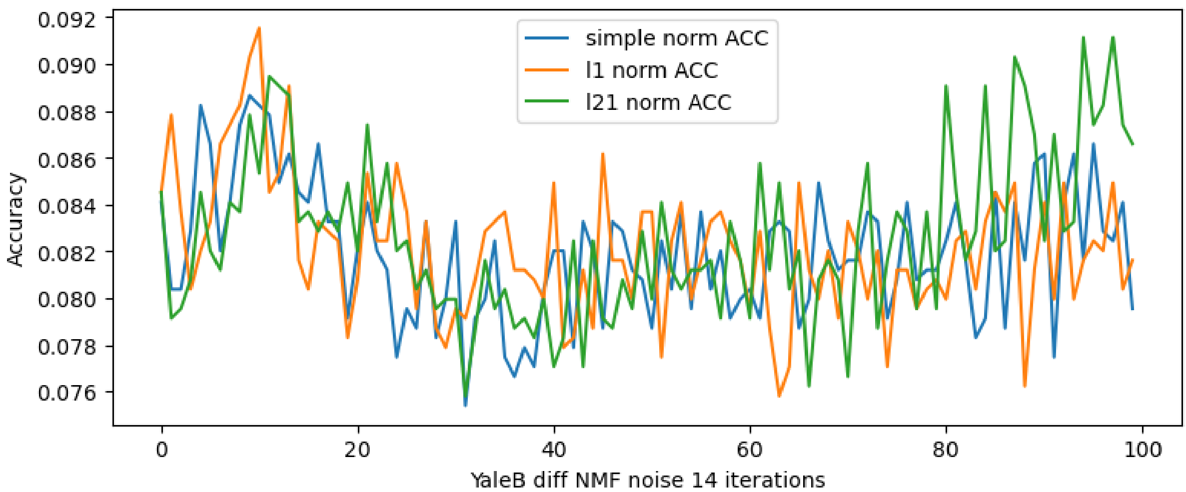

simple norm NMF
l1 norm NMF
l21 norm NMF
RMSE
ORL diff NMF no noise iterations

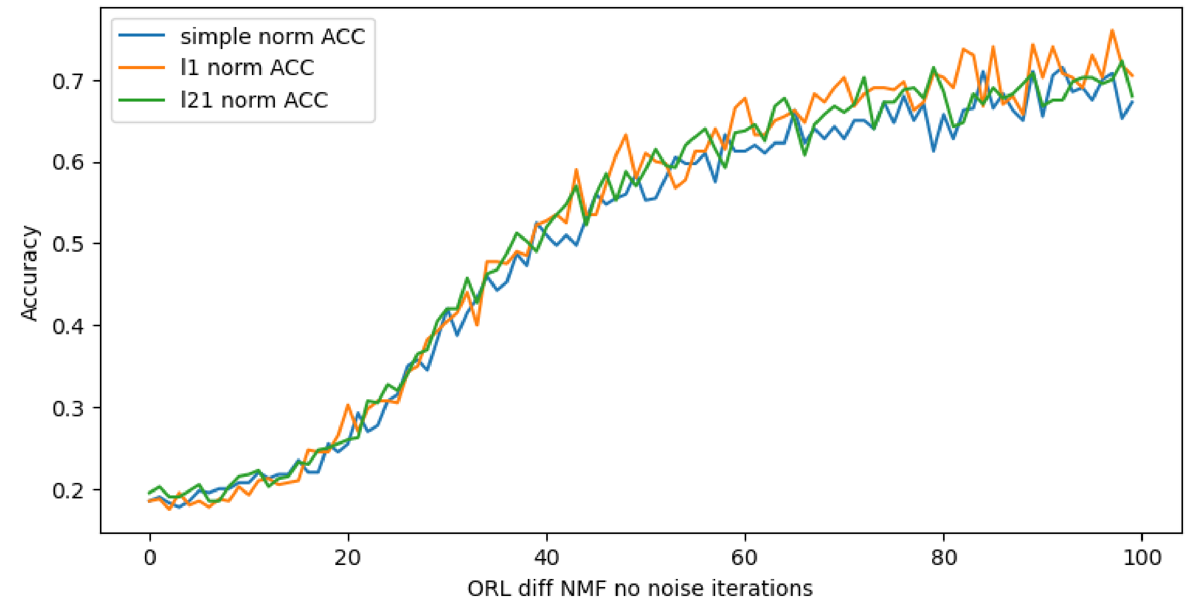

simple norm NMF
l1 norm NMF
l21 norm NMF
RMSE
ORL diff NMF noise 10 iterations

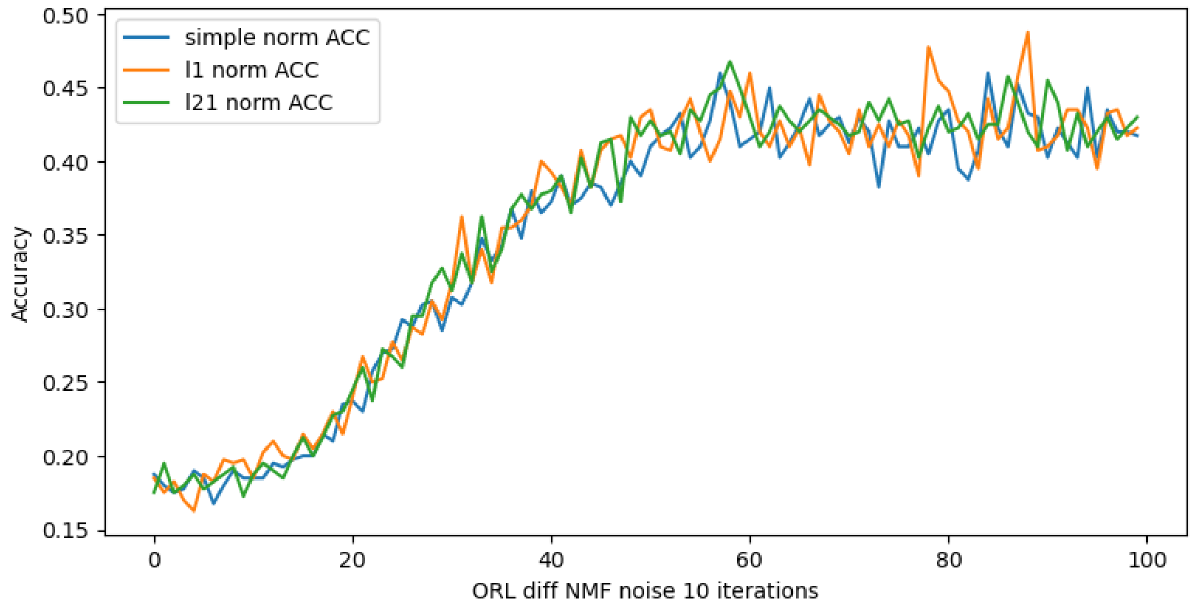

simple norm NMF
l1 norm NMF
l21 norm NMF
RMSE
ORL diff NMF noise 14 iterations

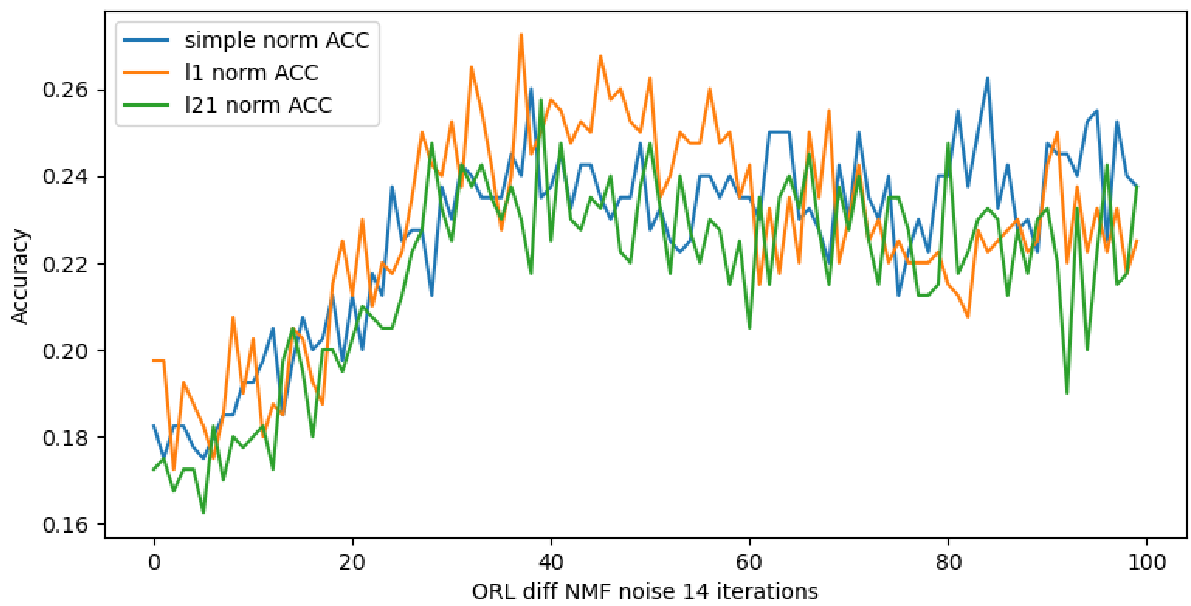

simple norm ACC
l1 norm ACC
l21 norm ACC
Accuracy
YaleB diff NMF no noise iterations

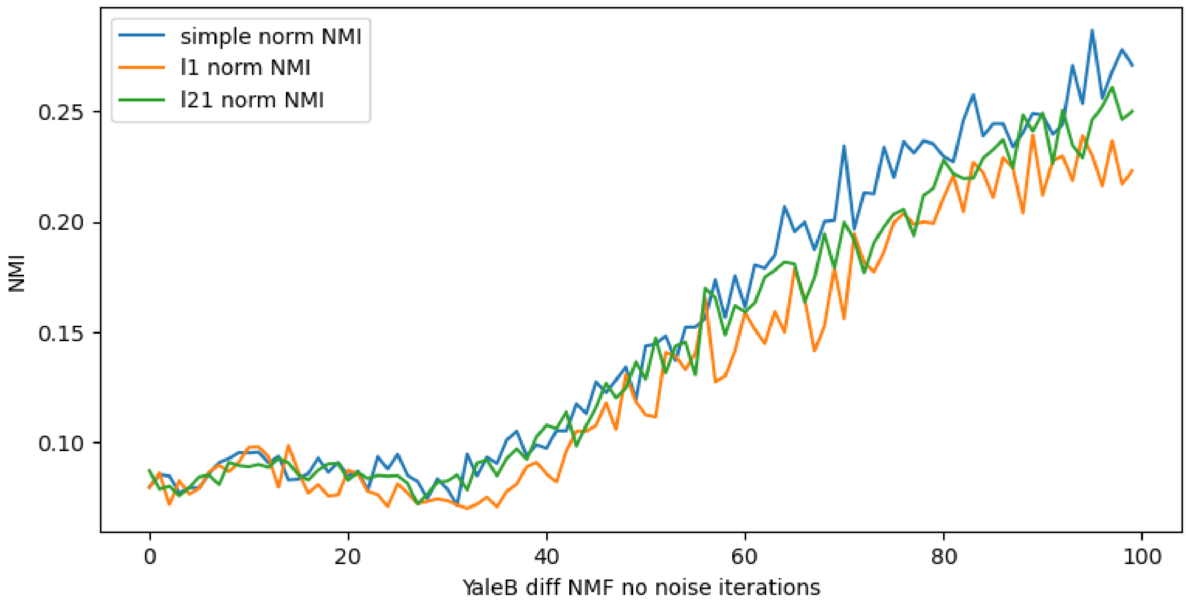

simple norm ACC
l1 norm ACC
l21 norm ACC
Accuracy
YaleB diff NMF noise 10 iterations

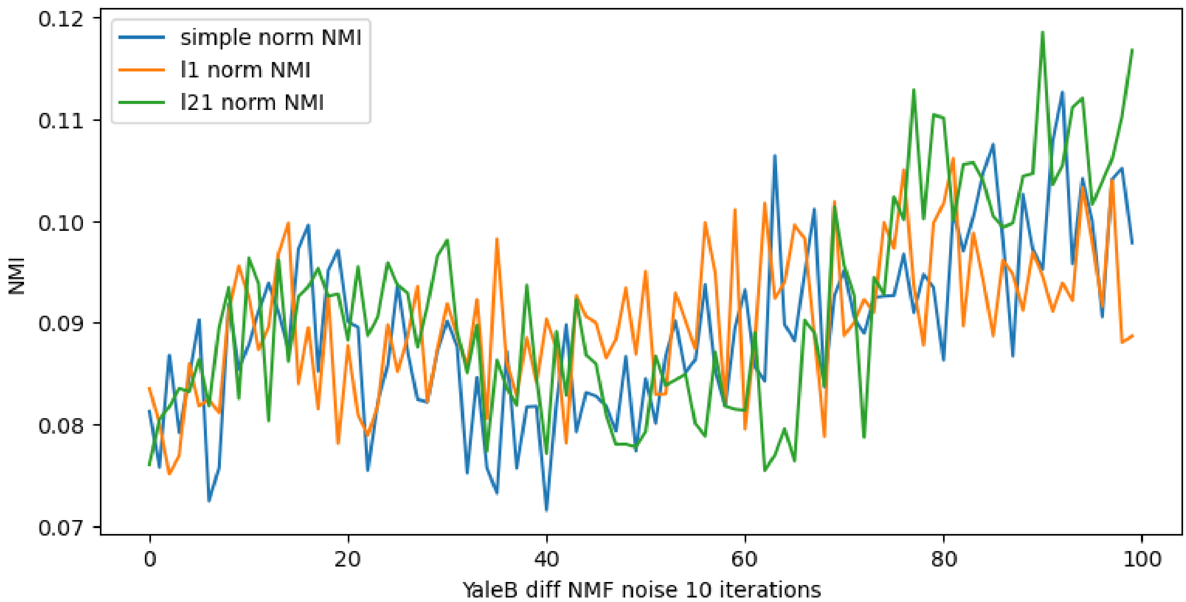

simple norm ACC
l1 norm ACC
l21 norm ACC
Accuracy
YaleB diff NMF noise 14 iterations

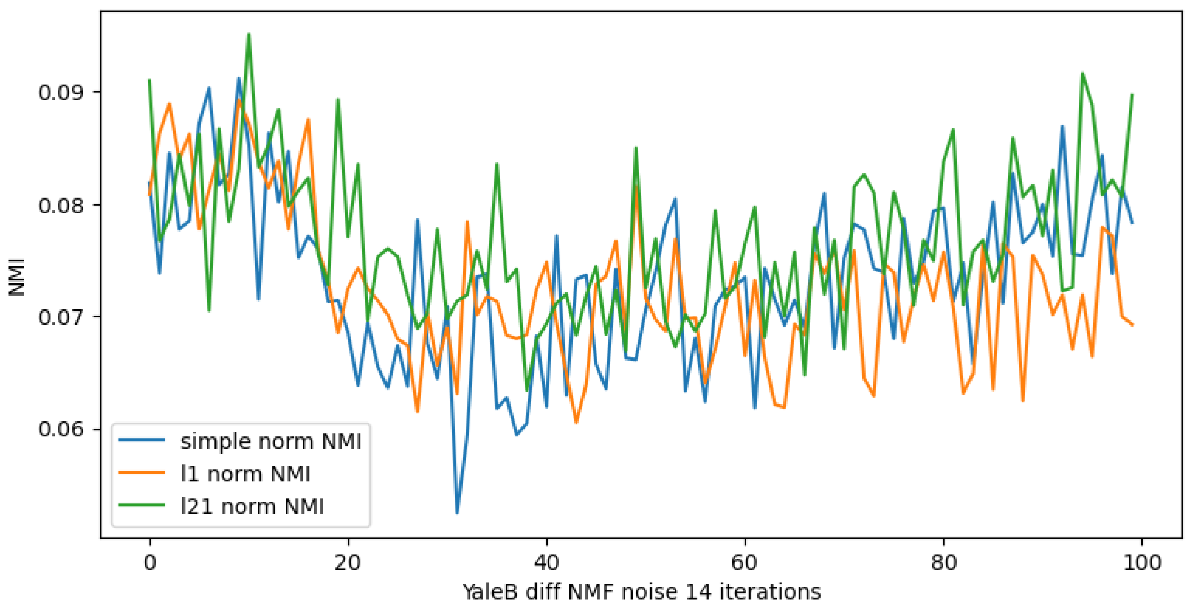

simple norm ACC
l1 norm ACC
l21 norm ACC
Accuracy
ORL diff NMF no noise iterations

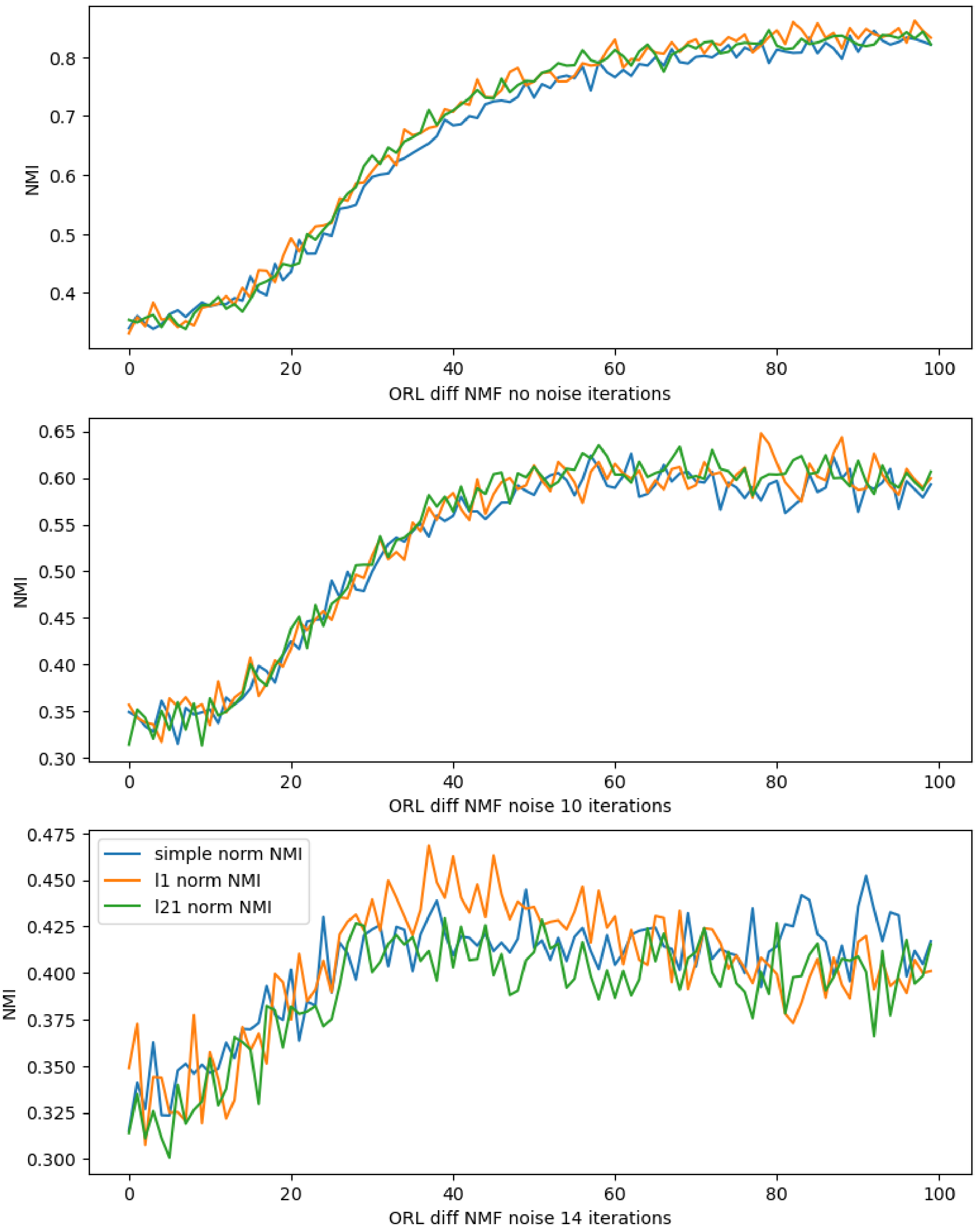

simple norm ACC
l1 norm ACC
l21 norm ACC
Accuracy
ORL diff NMF noise 10 iterations


simple norm ACC
l1 norm ACC
l21 norm ACC
Accuracy
ORL diff NMF noise 14 iterations


simple norm NMI
l1 norm NMI
l21 norm NMI
0.25
0.20
0.15
0.10
NMI
0
20
40
60
80
100
YaleB diff NMF no noise iterations


simple norm NMI
l1 norm NMI
l21 norm NMI
0.12
0.11
0.10
0.09
0.08
0.07
NMI
0
20
40
60
80
100
YaleB diff NMF noise 10 iterations


simple norm NMI
l1 norm NMI
l21 norm NMI
0.09
0.08
0.07
0.06
NMI
0
20
40
60
80
100
YaleB diff NMF noise 14 iterations


NMI
ORL diff NMF no noise iterations
NMI
ORL diff NMF noise 10 iterations
simple norm NMI
l1 norm NMI
l21 norm NMI
NMI
ORL diff NMF noise 14 iterations