# EE 779 Advanced Topics in Signal Processing
# NMF for Face Recognition
# Assigned: 24/10/16, Due: 04/11/16
# Indian Institute of Technology Bombay

## Note

- Before starting this, please complete the PCA assignment.
- This is an individual assignment. Please post your doubts on moodle.
- Relevant files including the articles [1], [2] and [3] are posted in moodle.
- Code and data (`nmfpack`) were adapted from the code provided by [1].

**Description** The objective of this assignment is to understand the application of non-negative matrix factorization (NMF) in face recognition. We are more interested in the NMF representation than the recognition itself. You are provided with the Matlab code for NMF implementation based on [1]. This includes a data base of face images (orl database). This is same as the one used in the PCA assignment. You will have to understand the algorithm and test it with few sample images (e.g., your face image). You will have to submit a two-page (maximum) writeup of your analysis and observations. Follow the steps (similar to the ones you did in the PCA assignment) and get started on NMF.

1. Download the file `pca_nmf_proj.zip` and extract its contents into a folder `pca_nmf_proj`.
2. Go through the file `nmf_main.m` and understand how it implements the NMF algorithm discussed in [2], [1]. Read the comments and code to understand the `MAT` files it creates.
3. In the `nmf_main.m` provided, a subset of the images ($40 \times 5$ using `orl_train.m`) is used to obtain the bases and activations ($\mathbf{V} = \mathbf{WH}$). Let the number of bases ($r$ - rank of $W$) be 49. Write your own code and name it `nmf_<firstname>.m`. Call `nmf_main` first to 'learn' and then to 'show' the output. We will be using the Frobenius norm distance measure in NMF. Hence you can use this call:

   ```
   >> nmf_main('learn', 101)
   ```

   The NMF algorithm keeps running and updates the `MAT` file in the `results` directory. To stop the algorithm, use `ctrl-C` to stop the program.

   Next to see the results, use this call:

   ```
   >> nmf_main('show', 101)
   ```

   Verify the performance of the NMF implementation and record your observations for the following three cases.

   (a) Use one of the original images (and name it `orlImgTrain`) from the training images. Reconstruct this image using the obtained $\mathbf{W}$ and name it as `orlImgTrainEst` . Obtain the mean-square error `trainMSE` by comparing these two. Include this value and images in your report.

   (b) Use one of the test images (and name it `orlImgTest`) by using `orldata_test.m`. Reconstruct this image using the obtained $\mathbf{W}$ and name it as `orlImgTestEst` . Obtain the mean-square error `testMSE` by comparing these two. Include this value and images in your report.

   (c) Create an image of your face so that it matches the dimension of the data used in the `orl` data. i.e., $46 \times 56$ and grey-scale image. Crop it so that it only has your face and name this image `myImg`. Reconstruct this image using the obtained $\mathbf{W}$ and name it as `myImgEst` . Obtain the mean-square error `myImgMSE` by comparing these two. Include this value and images in your report.

Compare and comment on the results obtained for the three cases.

4. Repeat (3), but for $r = 25$ bases vectors and create a file similar to the one in (3) and name it `nmf2_<firstname>.m`.

5. Compare the results you obtained with NMF and PCA, for specific images (one or two images from the database). Does your observations agree with claims in [1], [3] ? Summarize your observations on performance of NMF.

### Matlab tips

1. One way to display a vector as a grayscale image in Matlab.

```
>> imshow(reshape(img, 46, 56), [min(img) max(img)])
```

### References

## References

[1] Patrik O Hoyer. Non-negative matrix factorization with sparseness constraints. *J. Machine Learning Research*, 5:1457–1469, Nov. 2004.

[2] D. Lee and S. Seung. Learning the parts of objects by nonnegative matrix factorization. *Nature*, 401:788–791, 1999.

[3] Stan Z Li, XinWen Hou, Hongjiang Zhang, and Qiansheng Cheng. Learning spatially localized, parts-based representation. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2001.*, volume 1, pages I–207–I–212 vol.1, 2001.