CS598 PS 2017 – PROBLEM SET 2

This problem set is due on October 6th. Scanned/handwritten PDFs will not be accepted. If you are a graduate student, you have to do all the problems; undergraduates get extra credit for completing problems marked as graduate problems. For each day that your submission is late your grade will drop by 10%. Try to do the problem set by yourself, it will help you with understanding all the material we cover in class. All code submission has to be in Python.

Problem 1. An audio features project

The problem set bundle contains the sound-file `vl1.wav`. This is the sound of a 80s synthesizer that used to be very hip. The drumbeat in this sound is composed out of three "instruments". Each one is distinct from the others in its spectral composition and you should be able to tell them apart by listening to it. Your computer however cannot (yet).

Obtain the magnitude of the spectrogram of this recording. Use a window size of 1024 points, an overlap ¾ of the window size and apply a Hamming window. To make the spectrogram easy to see as an image plot, display its square root instead. By looking at it you should be able to "see" the difference between the three instruments.

Let's make your computer discover the three instruments in the beat:

1. Perform PCA on the magnitude spectrogram and extract three components. The spectrum (frequency) axis will be the input dimensions axis, and each time slice of the spectrogram will be a sample. Your PCA components should be three "eigenspectra".
2. Now extract three ICA components (remember to do PCA beforehand, you cannot drop the dimensionality with ICA only).
3. Now extract three NMF components (no need for PCA this time).

Plot the resulting features and weights for each of the three above cases. Note that in the case of PCA these will be the eigenvectors of the covariance. In ICA they will be the columns of what is known as the *mixing matrix* (when you perform ICA using $\mathbf{y} = \mathbf{W}\mathbf{x}$, the mixing matrix is the inverse of $\mathbf{W}$). In NMF they will be the columns of the $\mathbf{W}$ matrix, assuming the decomposition spectrogram = $\mathbf{WH}$. What observations can you make? How do the results differ? Which ones make the most sense?

Problem 2. Handwritten digit features

The problem set bundle contains the data file `digits.mat`. Load the data from that file and look for variable `d`. It contains a handwritten digit collection of 28×28 images that is used for handwriting recognition benchmarks. The pixel data will be ordered as a matrix so that each column is an unwrapped digit image (therefore of 28×28 = 784 dimensions). To display the i^{th} data points you can do: `imshow(reshape(d[:,i],(28,28),'F'))`. Do so and make sure that the data is loaded correctly and does indeed contain digits.

Just as before, perform PCA, ICA and NMF. Drop the dimensionality to 36 dimensions for all cases. Plot the resulting features from each transform as images (use the one-liner above for the eigenvectors, mixing matrix columns, etc). Make observations on how and why they differ.

Problem 3 (Graduate students). The geometry of handwritten digits

In the `digits.mat` file there is also another variable, vector `l`, that contains the digit label corresponding to each column of the matrix `d`.

Select only the columns that correspond to the digit 6. Perform PCA and drop the dimensionality down to two dimensions. Instead of plotting the features like we did above, plot the weights as a 2D scatter plot. Instead of dots display a small image of the digit corresponding to every point in the scatter plot. Make observations on how the shape of the shown digits changes as it is distributed in 2D space. Redo this experiment using the embedding from Laplacian Eigenmaps. For the computation of the distance matrix/graph use only the 10 nearest neighbors. Plot the resulting embedding as before. What's different now?

If you are bored and have nothing to do, try the same thing with other manifold algorithms and show the results.

Some notes on the code for the coding problems

With some easy Google searches you can find existing code to compute spectrograms, PCA, ICA, NMF, Laplacian Eigenmaps, etc. Feel free to use such code if you like, but do make a note in your report of which code you used (however, I do expect all of you to implement PCA yourself, make use of eigendecomposition or SVD routines for that). Of course, the more of the required code you write yourself the more extra credit you will get, and the more you will learn. Any code you write yourself, append it in your problem set submission.

If you do use code you find online keep in mind a couple of things:

1. In this class we talk of $M \times N$ data matrices as N samples of M dimensions. Some people prefer to think of such matrices as M samples of N dimensions. Make sure that you understand how the toolboxes you use treat their data and transpose your matrices appropriately.

2. As we talked in class many algorithms for the required decompositions can be iterative. This means that they will sometimes get stuck in local optima and produce poor results. Make sure you run your experiments 3-4 times to ensure that the data you obtain is consistent.

Good luck!