ENGR-E 511 Fall 2018: Assignment #3

Due on Sunday, October 21, 11:59P

Professor Minje Kim

Abhilash Kuhikar (akuhikar@iu.edu)

October 21, 2018

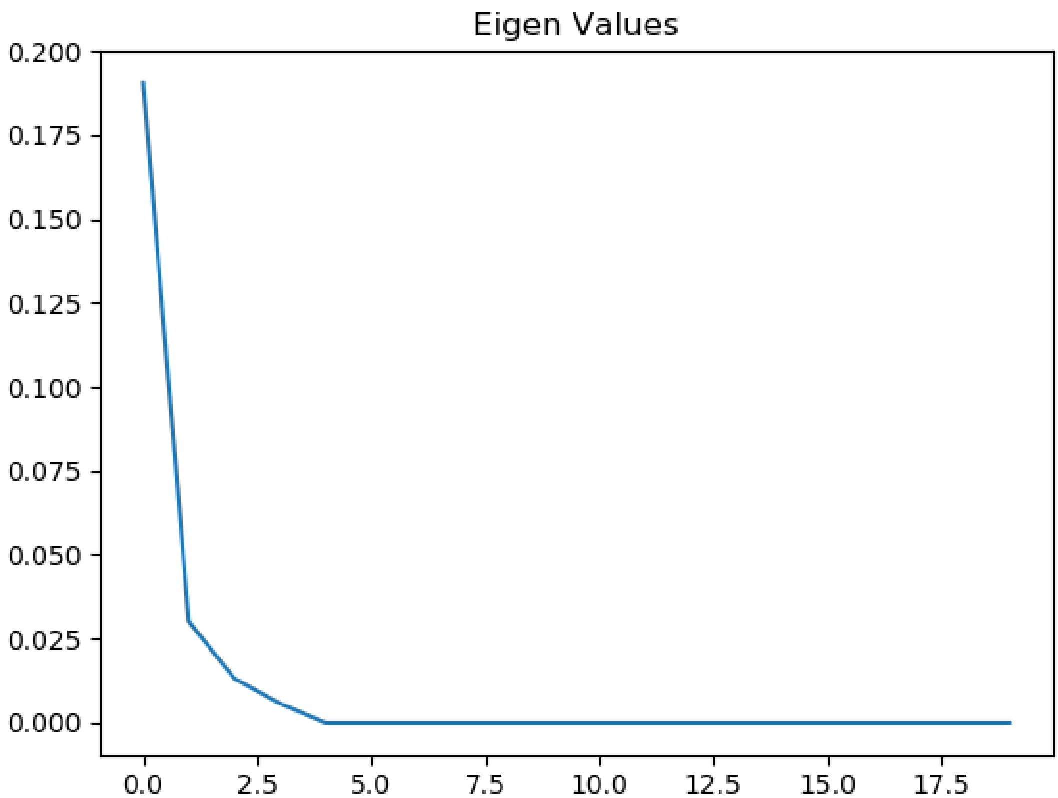
Contents

Problem 1: Instantaneous Source Separation

The objective is to use PCA and ICA to separate the sources using the 20 input sources. We can say that these 20 sources are the mixture of the original K sources mixed by a mixing matrix of 20 x K We will use ICA to separate the sources.

(a) **The first step is to reduce the dimensions using PCA**:
By doing PCA on the data, we can find that the first 4 eigenvalues are significantly larger than the rest of the eigenvalues.

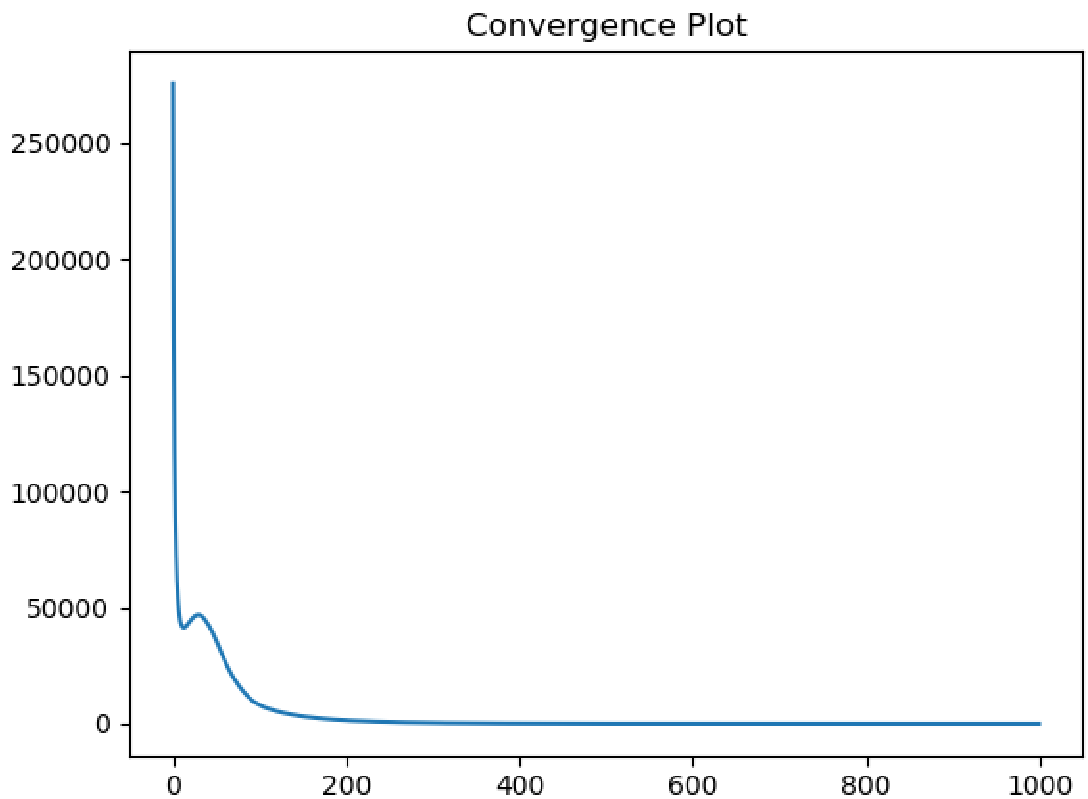
Eigenvalues:



Hence selecting first 4 eigenvectors which will correspond to the 4 independent sources.

(b) **Whiten the data in new reduced dimensions and find the rotation matrix using ICA**

The learning rate I used is **0.000001** and the number of iterations is 1000. Here's the convergence graph for the algorithm with this learning rate.



(c) It takes about 17 seconds for this algorithm to run. The 4 separates sources are clear and distinct from each other. If I increase the learning rate I found that the algorithm was not converging and instead was taking bigger jumps and was diverging. Looking at the convergence graph we can see that the ΔW converges nearly to zero after about 350-400 iterations.

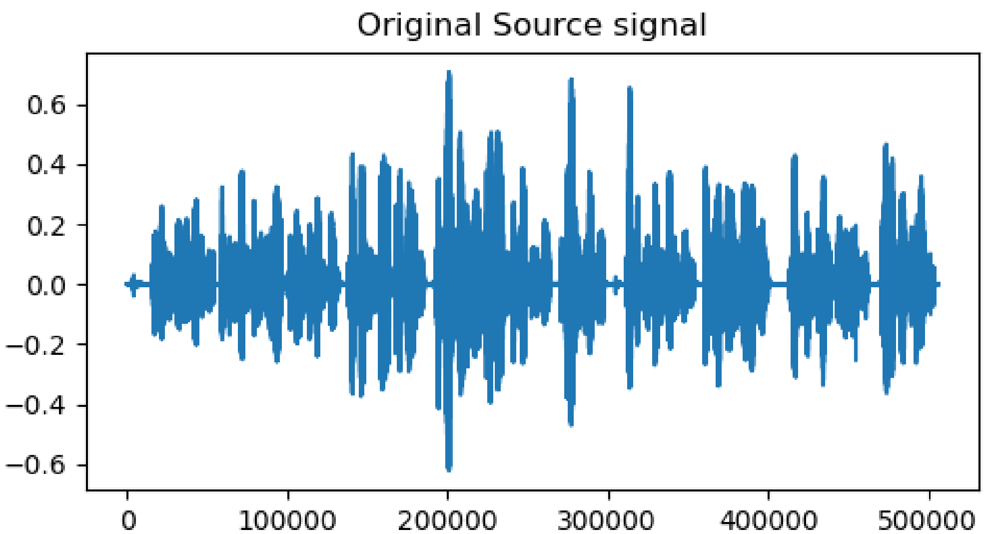
Problem 2: Single-channel Source Separation

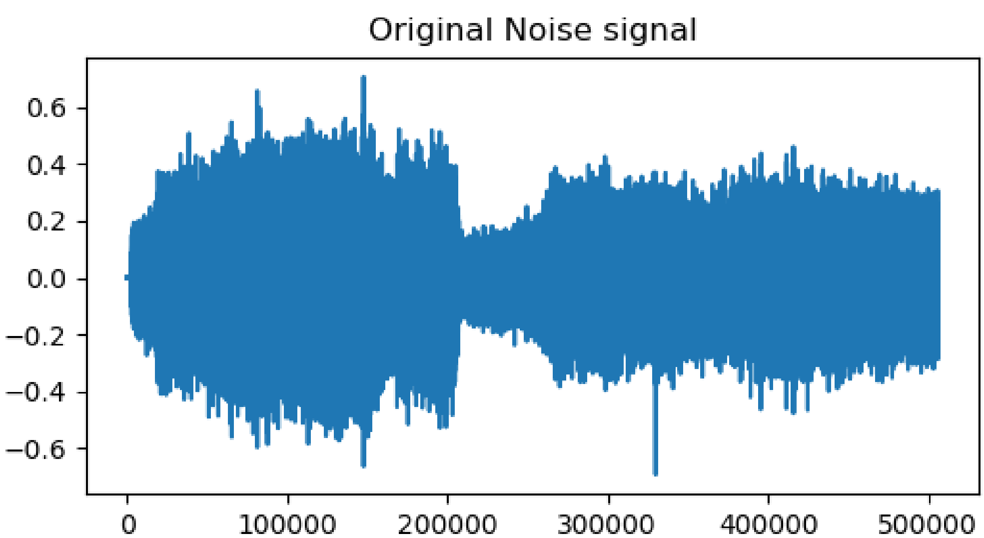
The objective is to use NMF to separate the single-channel source.

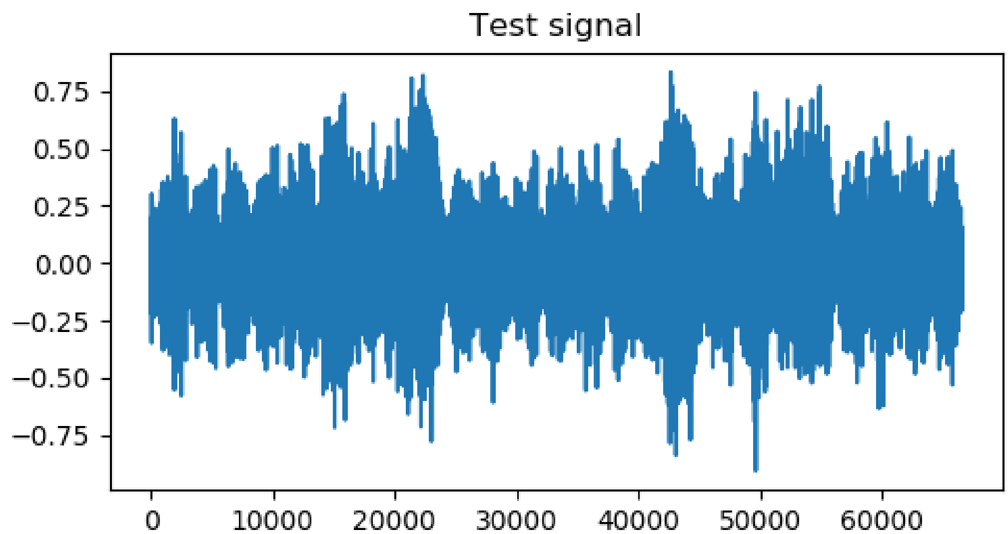
(a) Learn W_S using the update rules.

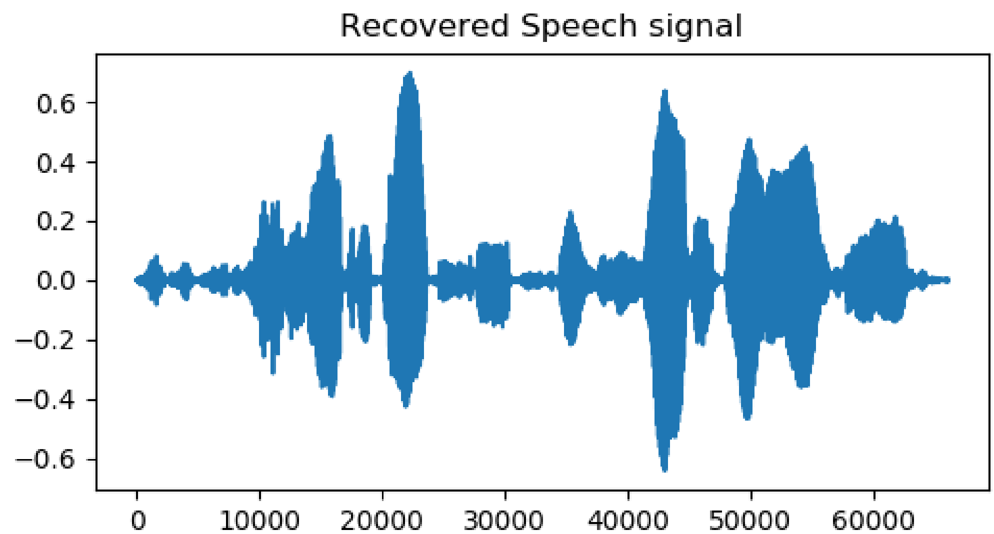
(b) Learn W_N using the update rules.

(c) Find the original source signal, noise signal, test signal and the recovered speech signal below









W_S and W_N are trained after 2000 iterations of update rules.
S nearly becomes equal to $W_S H_S$ N nearly becomes equal to $W_N H_N$

It takes about 40-50 seconds for the code to run and we can see from the images above that NMF was able to suppress the noise from the test signal and recover the speech.

Problem 3: Motor Imagery

The objective is to use classify the eeg samples into two classes by k nearest neighbors using locality sensitive hashing.

(a) First create an STFT matrix with frame size = 64 and hop size = 48 using a blackman window

(b) Take the first half of the spectogram for all the three channels for a given data sample.

(c) Select the 3rd to 7th rows from all the 3 channels and vecorize it to form a single vector of size 225 x 1

(d) kNN
- calculate hamming distance for a test sample with all the training samples
- take the nearest k neighbors based on these distance values
- see the labels for these neighbors from y_train
- count the number of labels for each class from these neighbors
- assign the class with the greater number of neighbors

(e) **Here are the accuracies for the different values of L and k**
Note : I have fixed(np.random.seed) the A matrix for a given value of L to study the accuracies.

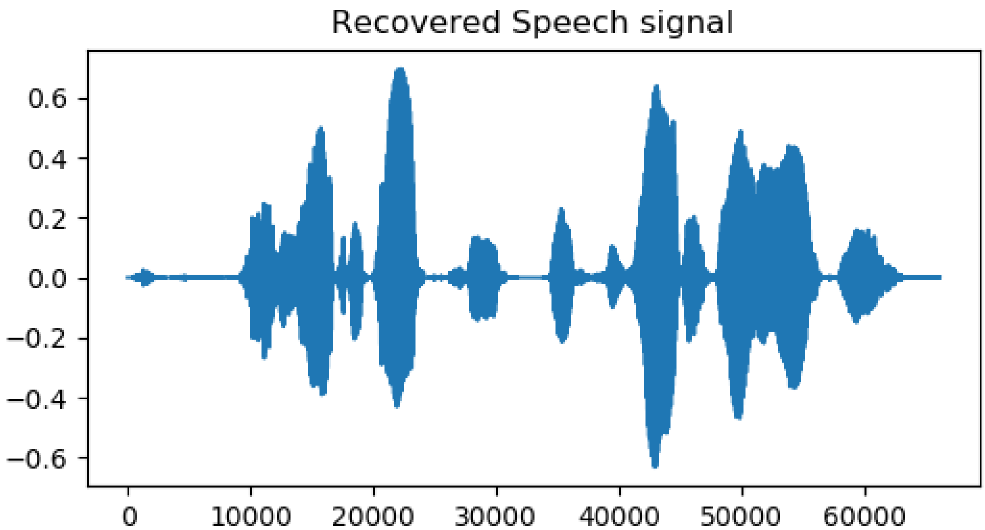
L	K	Accuracy
50	5	46.4%
50	10	50%
50	15	50%
100	5	64.28%
100	10	57.14%
100	15	64.28%
100	50	82.14%
150	5	57.14%
150	10	53.57%
150	20	42.85%
150	50	46.42%
200	5	85.71%
200	10	82.14%
200	20	85.71%
200	80	78.57%

We can see from the above table that the accuracy fluctuates generally between 45% and 75%. For a large value of L (e.g. 200) we can see that the accuracy is very high about 70-80 % for various values of k(neighbors). We can see that for some combinations of k and L, the accuracy is really good

Problem 4: kNN Source Separation

The objective is to use kNN to separate the sources and reduce the noise from test signal.
I have set $k = 25$

Let's look at the recovered signal using kNN :



We can see that the recovered signal in this case is a bit less noisier than the recovered signal in problem 2

References