Machine Learning I – Exercise Sheet 4

Jan Zwank

1 Cocktail party for two

We investigate the mixed signals contained in *toydata.txt* and *audiodata.txt*. A sample of these datapoints and their joint distribution are displayed in Figure 1 and 1.

We perform PCA and ICA on both datasets. The resulting components are also depicted in the aforementioned figures. Projecting onto these components yields the data depicted in Figure 3 to 6.

If we consider the figures corresponding to the toydata, it is easy to see that PCA selects basis vectors that maximize the variance in this direction. Since the toydata seems is distributed in a parallelogram this means that the basis vectors correspond to the directions given by opposing corners.

ICA on the other hand selects directions with the least statistical dependence. Therefore the basis vectors correspond to directions given by adjacent corners of the parallelogram.

For the audiodata, PCA and ICA follow the same premise, however the differences are less obvious due to the original distribution of the data.

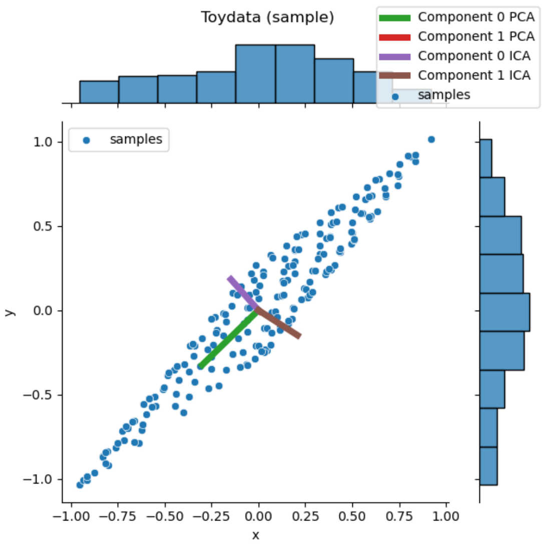

Fig. 1: A sample of datapoints from toydata.txt together with the PCA and ICA components.

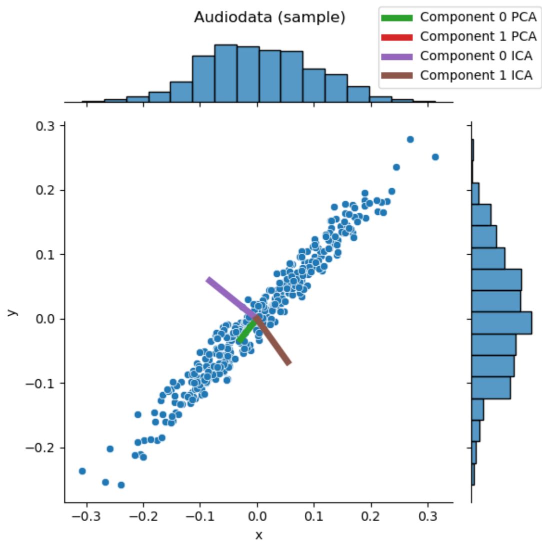

Fig. 2: A sample of datapoints from audiodata.txt together with the PCA and ICA components.

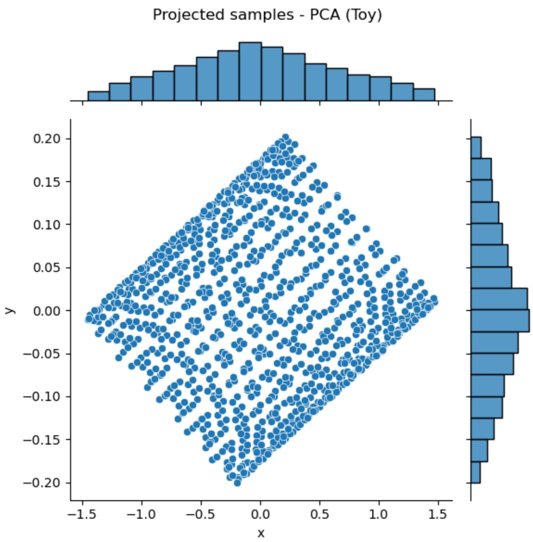

Fig. 3: Projected data using PCA on toy-data.txt

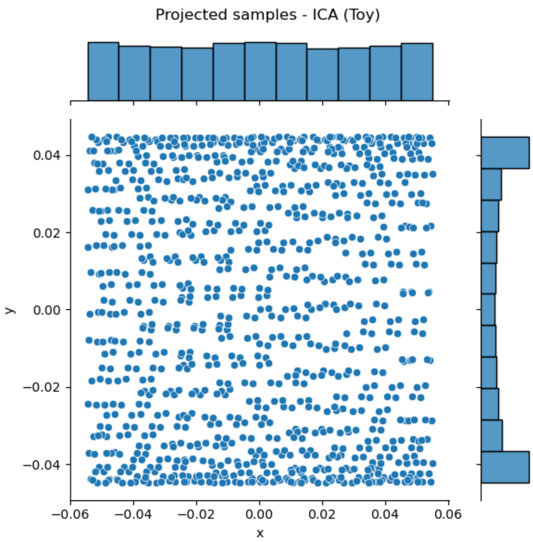

Fig. 4: Projected data using ICA on toy-data.txt

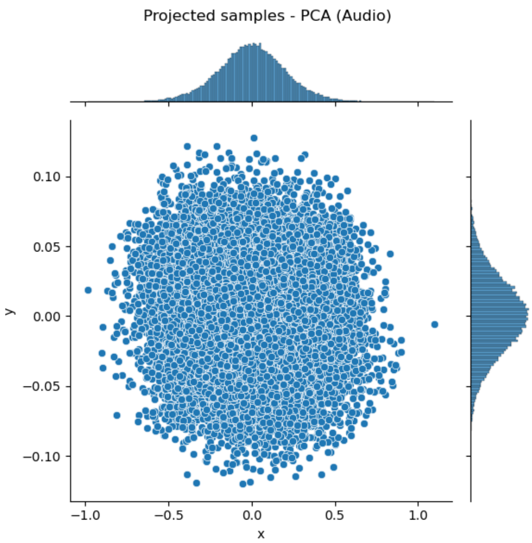

Fig. 5: Projected data using PCA on audiodata.txt

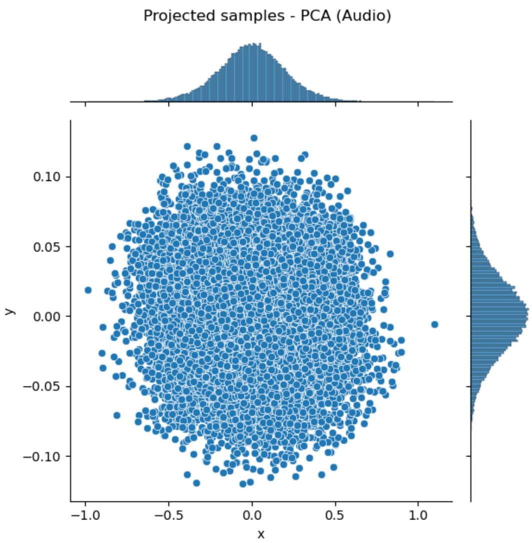

Fig. 6: Projected data using ICA on audiodata.txt

2 Ghostfaces

We use the dataset of grayscale images of faces from `http://conradsanderson.id.au/lfwcrop/lfwcrop_grey.zip`. For this exercise we use a subset of 1000 faces. In Figure 7 the first 20 such images are portrayed and in Figure 8 the average of all 1000 faces is displayed. The images can be interpreted as a 64×64 matrix. We flatten these matrices

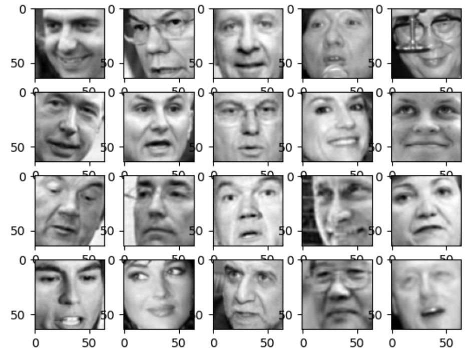

Fig. 7: A sample of 20 faces.

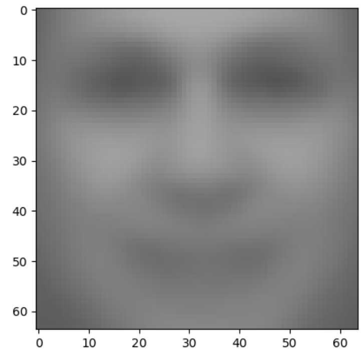

Fig. 8: The mean face over 1000 faces.

and store them as a 1000×4096 matrix. On this matrix we can now perform non-negative matrix factorization, resulting in 20 eigenfaces. These are depicted in Figure 9. The same procedure was performed with PCA (see Figure 10) and ICA (see Figure 11).

When we compare the eigenfaced of NMF with those of PCA and ICA, we see that NMF focuses more on certain areas of the faces that are more or less bright, whereas PCA and ICA focus more on the position and shape of certain aspects of the faces, such as noses and mouths.

Fig. 9: Eigenfaces for NMF.

Fig. 10: Eigenfaces for PCA.

Fig. 11: Eigenfaces for ICA.

3 Empirical Mode Decomposition

We load the data from *ex3_ signals.txt*, which contains two signals consisting of a mixture of 3 sine waves with different frequencies.

A plot of the first 500 points of the two signals is provided in Figure 12

We begin by performing ICA to estimate 2 source components (using `FastICA` from `sklearn.decomposition`[1]). They are displayed in Figure 13

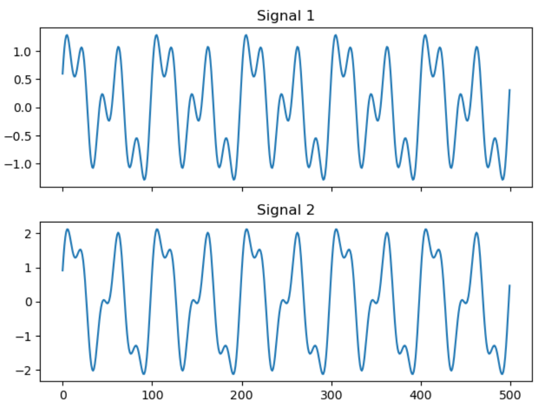

Fig. 12: The two signals from *ex3_ signals.txt*

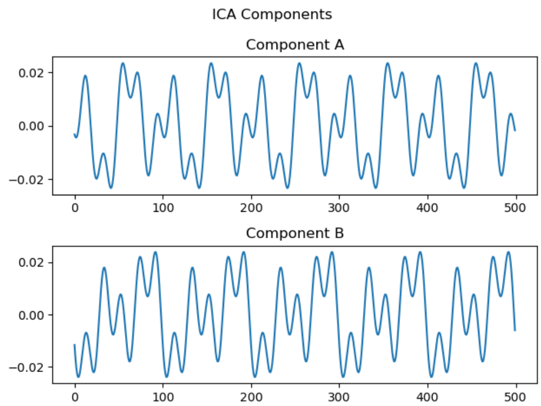

Fig. 13: Independent components from Signals (ICA)

We continue by performing empirical mode decomposition (using `PyEMD.EMD`) on Signal 1. The resulting intrinsic mode functions are displayed in Figure 14. We see that EMD correctly predicts three sine curves (the last three curves are only corner effects). ICA however fails to unmix the three sine signals because there are 3 sine signals, but only two ICA components.

We apply EMD again after adding some guassian noise with zero mean and a standard deviation of 0.1. The resulting intrinsic mode functions are included in Figure 15. The resulting intrinsic mode functions show that mode mixing in several places (especially in rows 2-4). We conclude that EMD is ill-suited to deal with noisy data.

We apply EEMD to the noisy data with noise widths 0.05, 0.1, 0.5, 1 and 2. Using this method solves the problem of mode mixing. The best results were obtained with `noise_width=0.05`. The corresponding intrinsic mode functions are provided in **??**.

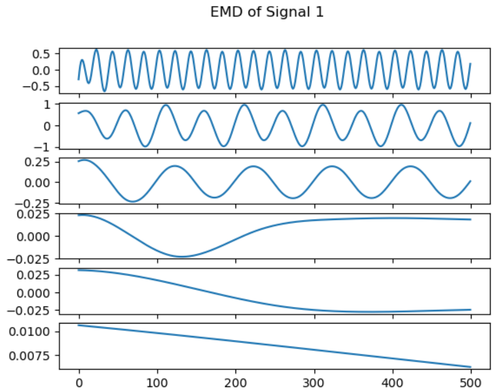

Fig. 14: Intrinsic mode functions of Signal 1 using EMD

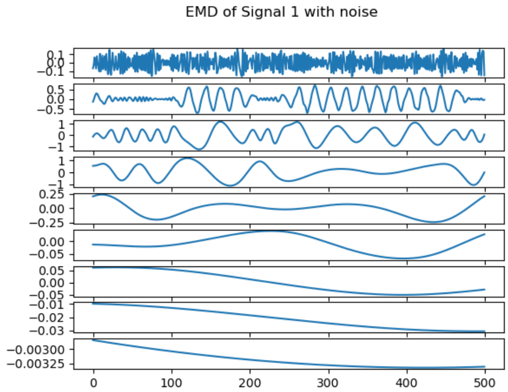

Fig. 15: Intrinsic mode functions of Signal 1 with added noise using EMD

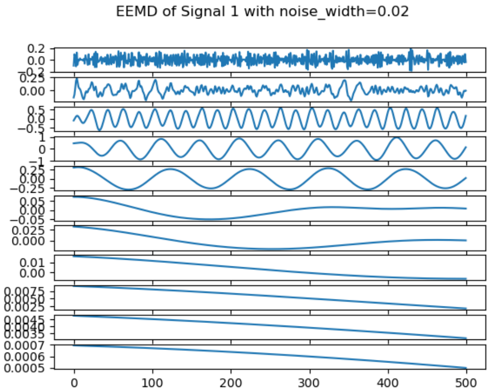

Fig. 16: EEMD with noise_w idht=0.02

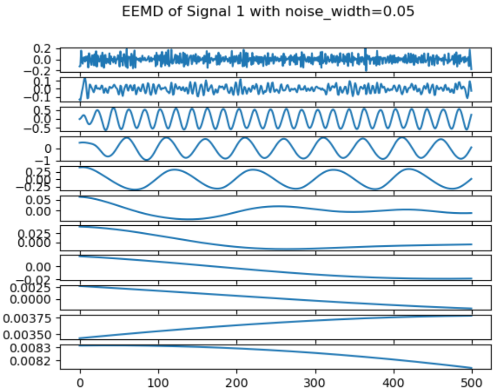

Fig. 17: EEMD with noise_widht=0.05

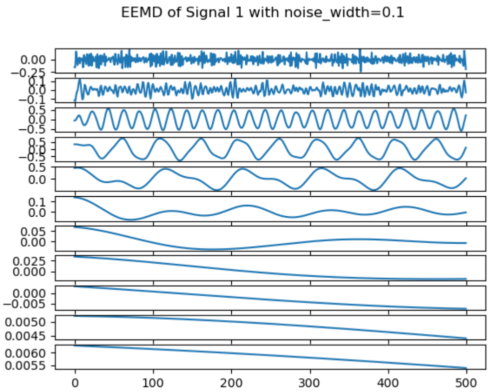

Fig. 18: EEMD with noise_widht=0.1

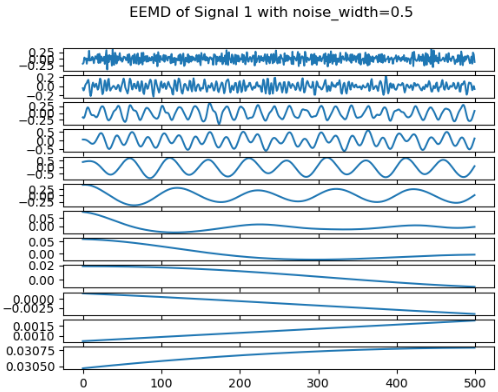

Fig. 19: EEMD with noise_widht=0.5

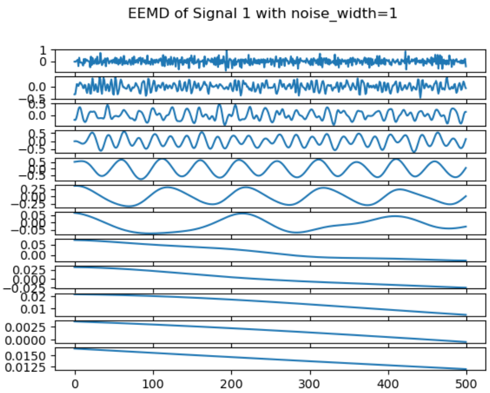

Fig. 20: EEMD with noise_widht=1

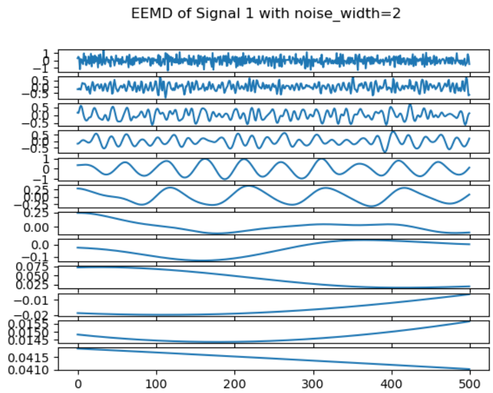

Fig. 21: EEMD with noise_widht=2