Marie LE GUILLY
Rasmus HAARSLEV
Troels THOMSEN

Artifact removal using PCA on EEG data

What are we trying to do?

When EEGs are recorded, electrodes are placed on the head of patients in order to measure the brain's electrical activity. However, these measures can be disturbed by noise caused by muscular movements such as eyelid movement. This creates large spikes in amplitude of the recorded signal artifacts. Our purpose is to try to remove them.

How do we remove artifacts?

We use PCA to identify components with an unusually large eigenvalues. If the eigenvalue is above the threshold found during calibration, there should be an artifact in that component and we try to remove it. We work with EEG data contained in a matrix X. We divide this matrix into several windows.

A window is a matrix $Y_{j,t} = X_{j,t}$ which contains the EEG data observed in the time interval between $t1$ and $t2$.

Y is of size J (Number of channels) $\times T$ (number of time points in window $T = t2 - t1 + 1$).

We normalize the data by subtracting the mean:

$$\mu_j = \frac{1}{T} \times \sum_t Y_{j,t}$$

$Z_{j,t} = Y_{j,t} - \mu_j$

Then we form the covariance matrix:

$$C_{j,k} = \frac{1}{T} \times \sum_t Z_{j,t} \times Z_{k,t}$$

We get the eigenvectors in matrix V and their eigenvalues in matrix E: $[E,V] = eig(C)$

Eigenvectors are then sorted according to their eigenvalues, with the largest one first in V.

The K largest ones are removed, based on a given threshold such that $\forall i < K,\ E_i > threshold$.

We project the data back onto the remaining eigenvectors:

$R_{q,t} = \sum_t V_{j,q} \times Z_{j,t}$ with $j = (K+1):J$

Finally we can project the eigenvectors back into the channel space:

$$Z_{reconstructed\,j,t} = \sum_q V_{j,q} \times R_{q,t}$$

And we add the mean back:

$$Y_{reconstructed\,j,t} = \mu_j + Z_{reconstructed\,j,t}$$

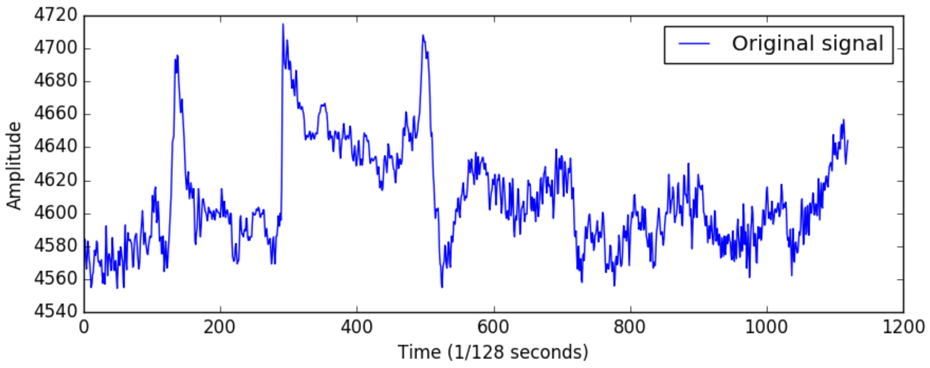

Figure 1: representation of one channel of an EEG dataset

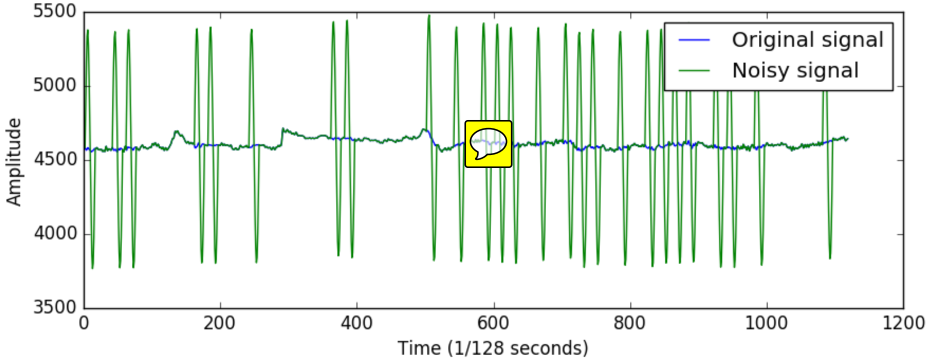

Figure 2: original dataset with simulated artifacts added

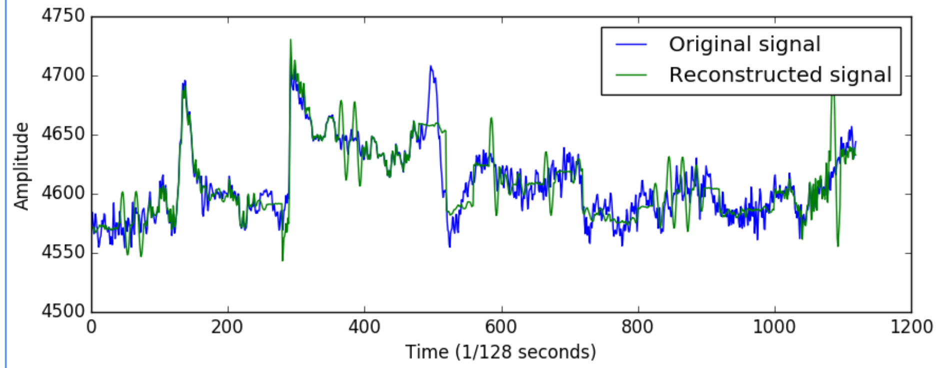

Figure 3: original dataset together with the dataset reconstructed from artifact dataset

Results

To compute the threshold for rejecting components we use a clean dataset as a training set. We calibrate the threshold on the first 20% of the dataset and then add simulated artifacts to the rest of it. We use three different kind of threshold:

- The maximum eigenvalue found in the windows of the training set
- The average of the eigenvalues of the windows of the training set
- The average of the maximum of each window of the training set

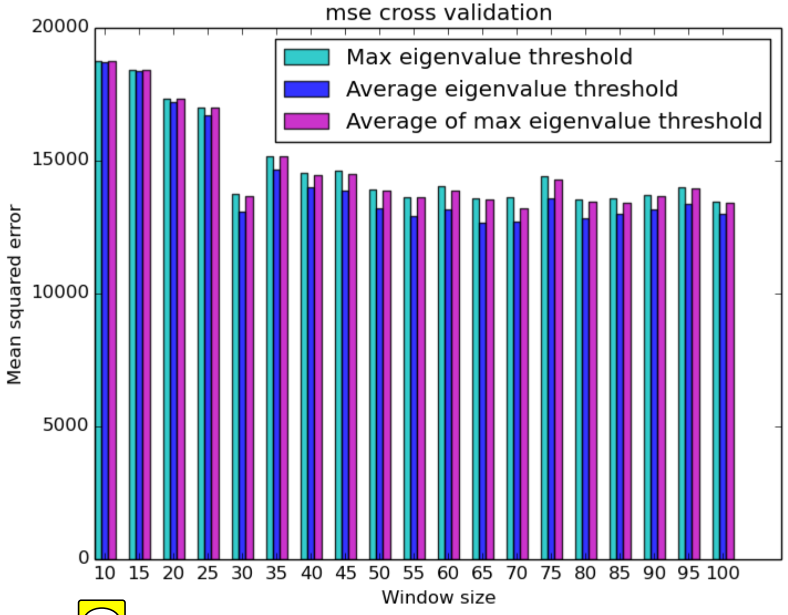

Figure 3: cross-validation between thresholds and window size

We can measure sensivity/specificity:

Threshold	Max	AVG	MAX_AVG
Sensitivity	1.0	1.0	1.0
Specificity	0.64	0.0	0.61

For windows of size 128 samples, we can remove artifacts at a rate of 35Hz.