Problem

Detrending in EDA SCR filtering invalidates the results of the zero crossing algorithm. This can lead to undetectability of peaks in some cases. In the following we will show why and how this happens.

We have a datasets that comprises of 27 recordings of a period of roughly 45 minutes of EDA data. For some of the datasets the SCR algorithm returns zero peaks. The following code was used on the signal in figure 1.

```
ts, filtered, onsets, peaks, amplitudes = eda(signal=eda_signal,
show=False, sampling_rate=1000)
print('Number of Onsets: ' + str(len(onsets)))
```

The resulting output is:

```
Number of Onsets: 0
```

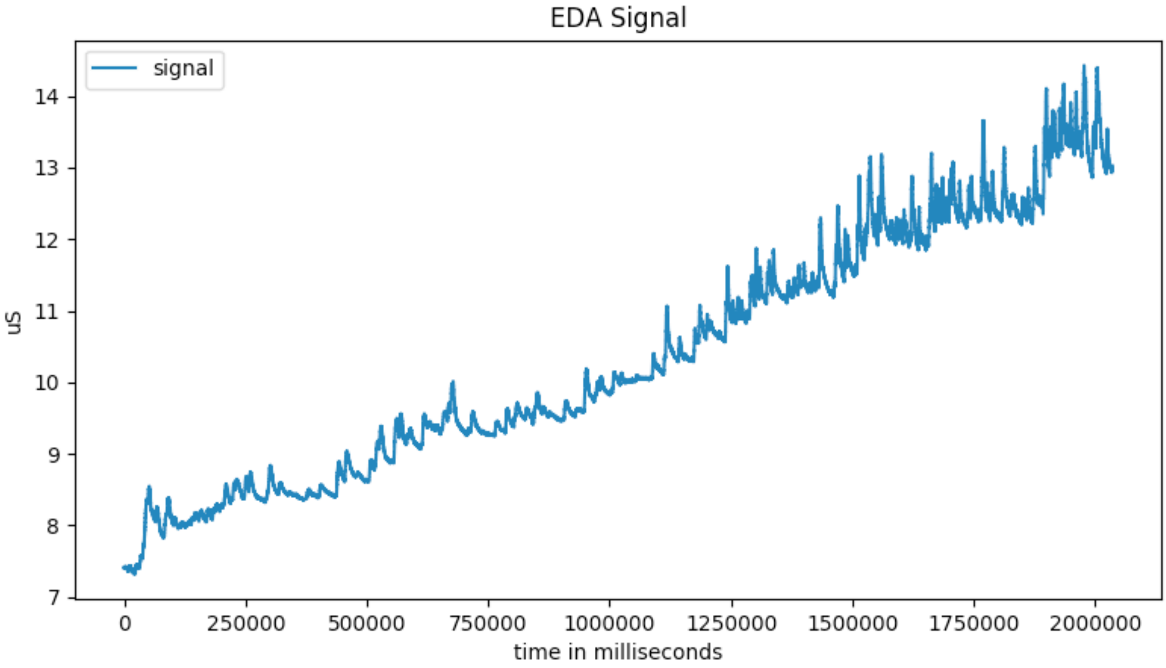

Figure 1: EDA signal from our dataset.

It seems wrong that there is no single peak in the signal. So we decided to take a look into the code. The zero crossings were determined by regarding the detrended derivative of the signal. In the kbk_scr() method the signal segment before the first zero crossing is evaluated. If their sum is greater than zero the first zero crossing is rejected. This is exactly where we think the issue lies. While the indices of the zero crossings were determined using the detrended derivative this evaluation of the first part is performed on the non detrended derivative. Figure 2 shows the discrepancy between derivative and detrended derivative.

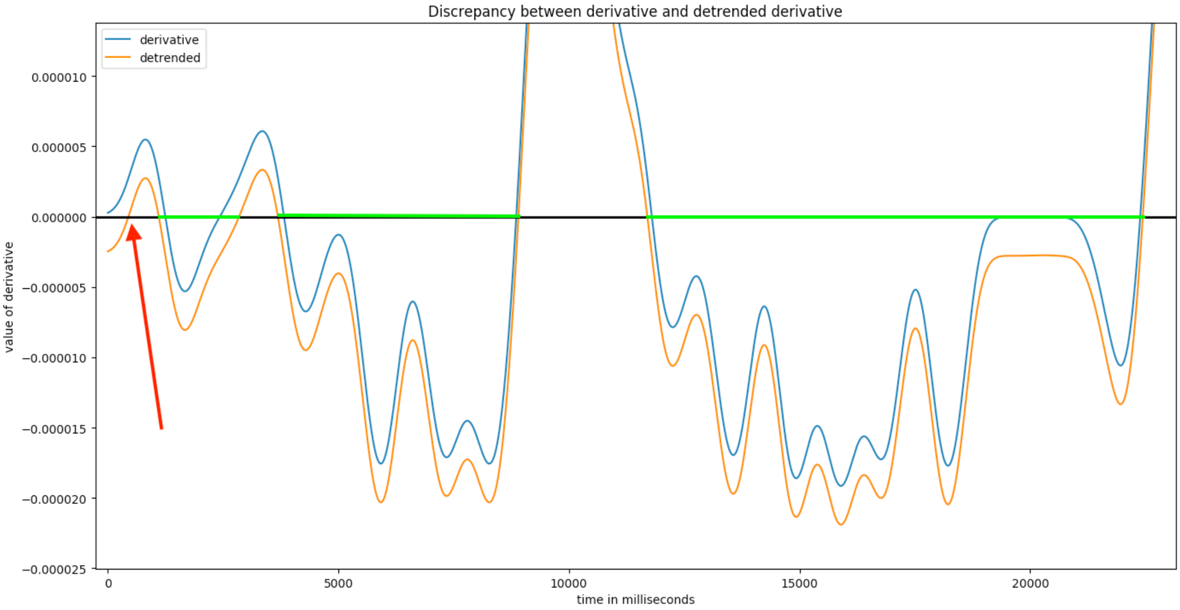

Figure 2: Discrepancy between derivative and detrended derivative.

For the derivative the total of the beginning of the signal is greater than zero which leads to the first zero crossing (red arrow) to be rejected. The derivative is then regarded in the intervals which are marked with the green lines. The threshold is determined on the non detrended derivative as well. Now the algorithm tries to find the highest values in the green intervals and rejects all of them as they by far smaller than 10% of the threshold. In figure 3 the results for zero crossing detection without detrending is highlighted. Again the red arrow shows the rejected zero crossing.

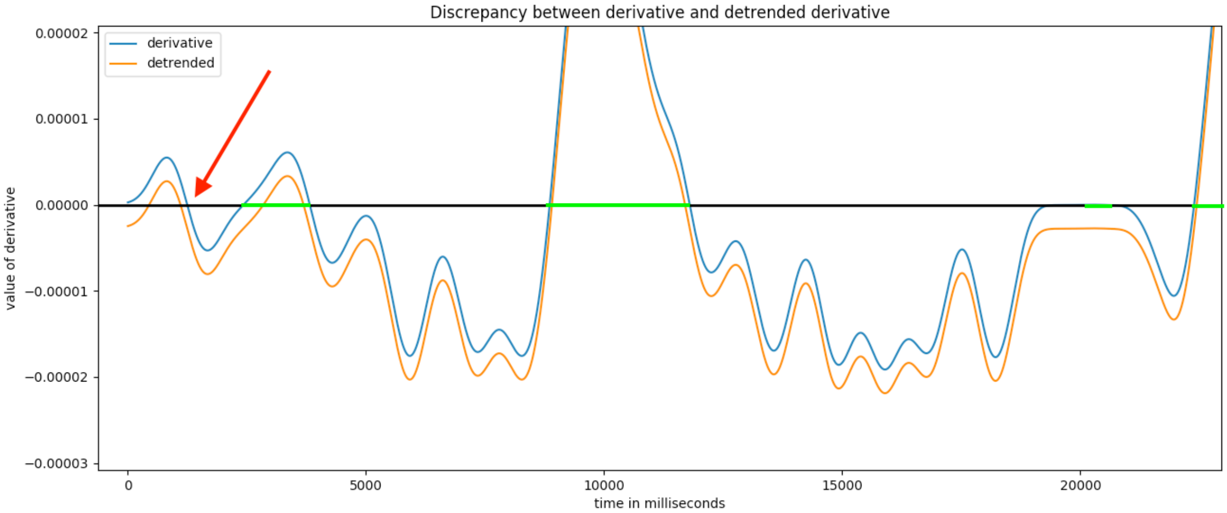

Figure 3: Discrepancy between derivative and detrended derivative. Highlighted are the results for zero crossing detection without detrending.

We conclude that detrending might not effect the derivative significantly. Still it slightly affects the positions of the zero crossings. This minor difference in the derivatives leads to wrong processing of the indices as they were calculated on the detrended version while the pruning is performed on the non detrended version.

As a solution we propose to turn off detrending.