# PCA_for_FetalECG_solution

December 20, 2022

# 1 PCA for Source Separation of Abdominal ECG Signals

## 1.1 Introduction

In this exercise we use PCA for the separation of maternal and fetal electrocardiography (ECG) signals in abdominal ECG (aECG) data recorded on the belly of a pregnant woman. Due to the low signal strengh of fetal ECG (fECG) signals it is an "algorithmic challenge" to properly separate fECG from much stronger maternal ECG (mECG) signals [1].

The present example uses a simplified version of the method proposed by Varanini et al. [2].

## 1.2 References

[1] R. Kahankova et al., "A Review of Signal Processing Techniques for Non-Invasive Fetal Electrocardiography," IEEE Reviews in Biomedical Engineering, vol. 13, pp. 51–73, 2020, doi: 10.1109/RBME.2019.2938061.

[2] M. Varanini, G. Tartarisco, L. Billeci, A. Macerata, G. Pioggia, and R. Balocchi, "An efficient unsupervised fetal QRS complex detection from abdominal maternal ECG," Physiol. Meas., vol. 35, no. 8, pp. 1607–1619, Aug. 2014, doi: 10.1088/0967-3334/35/8/1607.

[3] Source of data: https://physionet.org/content/challenge-2013/1.0.0/

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy.signal import filtfilt, butter
from biosppy.signals import ecg
from sklearn.decomposition import PCA

import plotly.graph_objects as go
from plotly.offline import init_notebook_mode, iplot
from plotly.subplots import make_subplots
init_notebook_mode(connected=True)  # initiate notebook for offline plot

import sys
sys.path.append(r"..")
from mqrs_utils import cancel_mqrs
```

```python
[2]: # load abdominal ECG (aECG) data
     # transformed from initial source: https://physionet.org/content/challenge-2013/
      ↪1.0.0/set-a/a13.dat
     filename = '../aecg_a13.hdf5'
     aecg = pd.read_hdf(filename, key='signals').values
     fs = 1000
     t = np.arange(aecg.shape[0]) / fs
```

```python
[3]: # bandpass filter data
     b, a = butter(4, np.asarray([3, 45])/(fs/2), btype='bandpass')
     aecg = filtfilt(b, a, aecg.transpose()).transpose()

     # plot
     fig = go.Figure()
     for i in range(aecg.shape[1]):
         fig.add_trace(go.Scatter(x=t, y=aecg[:, i], name='AECG{:d}'.format(i)))
     fig.update_xaxes(title='Time (s)')
     fig.update_yaxes(title='AECG Amplitude (A.U.)')
     fig.update_layout(title='Bandpass-Filtered AECG Signals')
     fig.show()
```

```python
[4]: # apply first PCA for enhancing maternal ECG component
     pca1 = PCA()
     pc1 = pca1.fit_transform(aecg)

     # maternal ECG as the first principal component, note that this
     # remains a guess and would need to be automated in the final solution
     maternal_ecg = pc1[:, 0]

     # detect maternal QRS peaks
     ts, filtered, mqrs_peaks = ecg.ecg(maternal_ecg, fs, show=False)[:3]

     # plot
     fig = go.Figure()
     for i in range(pc1.shape[1]):
         fig.add_trace(go.Scatter(x=t, y=pc1[:, i], name='PC1[:,{:d}]'.format(i)))
         if i == 0:
             fig.add_trace(go.Scatter(x=t[mqrs_peaks], y=pc1[mqrs_peaks, i],␣
       ↪name='mQRS-Peaks',
                                      mode='markers', marker_color='red',␣
       ↪marker_symbol='circle-open'))
     fig.update_xaxes(title='Time (s)')
     fig.update_yaxes(title='PC1 (A.U.)')
     fig.update_layout(title='Principal Components of First PCA Used to Enhance mECG␣
       ↪Signal')
     fig.show()
```

```
[5]: # remove maternal QRS complexes from signal to obtain a best possible fetal ECG␣
     ↪signal
     x_residual, mecg_estimations = cancel_mqrs(fs, pc1, mqrs_peaks.astype(np.
     ↪float64))

     # plot
     fig = make_subplots(rows=3, cols=1, shared_xaxes=True)
     fig.add_trace(go.Scatter(x=t, y=pc1[:,0], name='Maternal ECG'), row=1, col=1)
     fig.add_trace(go.Scatter(x=t[mqrs_peaks], y=pc1[mqrs_peaks, 0],␣
     ↪name='mQRS-Peaks', marker_color='red',
                             legendgroup='mQRS', mode='markers',␣
     ↪marker_symbol='circle-open'), row=1, col=1)
     fig.add_trace(go.Scatter(x=t, y=mecg_estimations[:, 0], name='Interpolated mQRS␣
     ↪Signal'), row=2, col=1)
     fig.add_trace(go.Scatter(x=t[mqrs_peaks], y=mecg_estimations[mqrs_peaks, 0],␣
     ↪name='mQRS-Peaks', marker_color='red',
                             legendgroup='mQRS', showlegend=False, mode='markers',␣
     ↪marker_symbol='circle-open'), row=2, col=1)
     fig.add_trace(go.Scatter(x=t, y=x_residual[:, 0], name='mQRS-free Signal'),␣
     ↪row=3, col=1)
     fig.add_trace(go.Scatter(x=t[mqrs_peaks], y=x_residual[mqrs_peaks, 0],␣
     ↪name='mQRS-Peaks', marker_color='red',
                             legendgroup='mQRS', showlegend=False, mode='markers',␣
     ↪marker_symbol='circle-open'), row=3, col=1)
     fig.update_xaxes(title='Time (s)', row=3, col=1)
     fig.update_layout(title='Maternal QRS Cancellation')
     fig.show()
```

```
[6]: # apply second PCA for enhancing fetal ECG component in residual signal
     pca2 = PCA()
     pc2 = pca2.fit_transform(x_residual)

     # fetal ECG as the first principal component, note that this
     # remains a guess and would need to be automated in the final solution
     fetal_ecg = pc2[:, 0]
     # detect fetal QRS peaks
     ts, filtered, fqrs_peaks = ecg.ecg(fetal_ecg, fs, show=False)[:3]

     # plot
     fig = go.Figure()
     for i in range(pc2.shape[1]):
         fig.add_trace(go.Scatter(x=t, y=pc2[:, i], name='PC2[:,{:d}]'.format(i)))
         if i == 0:
             fig.add_trace(go.Scatter(x=t[fqrs_peaks], y=pc2[fqrs_peaks, i],␣
     ↪name='fQRS-Peaks',
```

```
                                      mode='markers', marker_color='black',␣
      ↪marker_symbol='triangle-down-open'))
fig.update_xaxes(title='Time (s)')
fig.update_yaxes(title='PC2 (A.U.)')
fig.update_layout(title='Principal Components of Second PCA Used to Enhance␣
      ↪fECG Signal')
fig.show()
```

```
[7]:  # plot for summarizing all
      fig = make_subplots(rows=2, cols=1, shared_xaxes=True)
      # maternal ECG with mQRS
      fig.add_trace(go.Scatter(x=t, y=maternal_ecg, name='Maternal ECG'), row=1,␣
        ↪col=1)
      fig.add_trace(go.Scatter(x=t[mqrs_peaks], y=maternal_ecg[mqrs_peaks],␣
        ↪name='mQRS-Peaks',
                               marker_color='red', mode='markers',␣
        ↪marker_symbol='circle-open'), row=1, col=1)
      # fetal ECG with fQRS
      fig.add_trace(go.Scatter(x=t, y=fetal_ecg, name='Fetal ECG'), row=2, col=1)
      fig.add_trace(go.Scatter(x=t[fqrs_peaks], y=fetal_ecg[fqrs_peaks],␣
        ↪name='fQRS-Peaks',
                               marker_color='black', mode='markers',␣
        ↪marker_symbol='triangle-down-open'), row=2, col=1)
      fig.update_xaxes(title='Time (s)', row=2, col=1)
      fig.update_layout(title='Maternal vs. Fetal ECG')
      fig.show()
```

```
[8]:  from IPython.display import display, Math, Latex
      display(Latex(r"\newpage"))
```

## 2   Exercise Questions

Please provide your answers directly below each question.

### 2.1   Question 1

Determine the maternal heart rate, both expressed in `Hz` and `beats/min`.

```
[9]: # estimate mqrs
     mrr = np.diff(mqrs_peaks) / fs
     mhr_median = 60 / np.median(mrr)
     mhr_mean = 60 / np.mean(mrr)

     print('Maternal HR: mean={:.1f} bpm; median={:.1f} bpm'.format(mhr_mean,␣
       ↪mhr_median))
     print('Maternal HR: mean={:.1f} Hz; median={:.1f} Hz\n'.format(mhr_mean/60,␣
       ↪mhr_median/60))
```

```
Maternal HR: mean=81.6 bpm; median=79.3 bpm
Maternal HR: mean=1.4 Hz; median=1.3 Hz
```

```
[10]: plt.plot(t[mqrs_peaks][:-1], 60/np.diff(t[mqrs_peaks]), label='Beat-to-Beat HR')
      plt.plot(t[[0, -1]], [60/np.median(np.diff(t[mqrs_peaks]))]*2, '--r',␣
        ↪label='Median HR')
      plt.plot(t[[0, -1]], [60/np.mean(np.diff(t[mqrs_peaks]))]*2, ':k', label='Mean␣
        ↪HR')
      plt.legend()
      plt.xlabel('Time (s)')
      plt.ylabel('Maternal Heart Rate (bpm)')
      plt.title('Maternal Heart Rate Estimation')
      plt.show()
```

Maternal Heart Rate Estimation
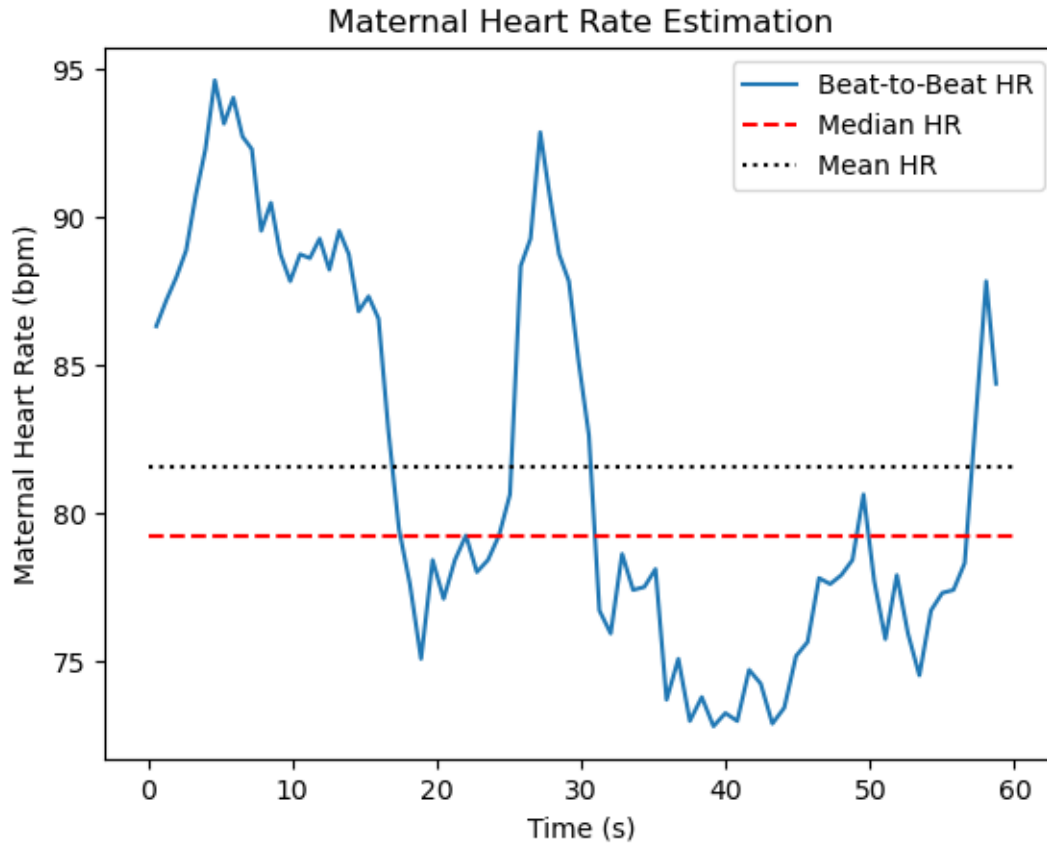
## 2.2 Question 2

Determine the fetal heart rate, both expressed in `Hz` and `beats/min`.

```
[11]:  # estimate fqrs
       frr = np.diff(fqrs_peaks) / fs
       fhr_median = 60 / np.median(frr)
       fhr_mean = 60 / np.mean(frr)

       print('Fetal HR: mean={:.1f} bpm; median={:.1f} bpm'.format(fhr_mean,␣
         ↪fhr_median))
       print('Fetal HR: mean={:.1f} Hz; median={:.1f} Hz\n'.format(fhr_mean/60,␣
         ↪fhr_median/60))
       print('Due to missing detections of certain fQRS peaks the the mean value is␣
         ↪underestimating the real HR.\nTherefore the median value shall be used.')
```

```
Fetal HR: mean=116.7 bpm; median=125.9 bpm
Fetal HR: mean=1.9 Hz; median=2.1 Hz

Due to missing detections of certain fQRS peaks the the mean value is
underestimating the real HR.
```

Therefore the median value shall be used.

```
[12]: plt.plot(t[fqrs_peaks][:-1], 60/np.diff(t[fqrs_peaks]), label='Beat-to-Beat HR')
      plt.plot(t[[0, -1]], [60/np.median(np.diff(t[fqrs_peaks]))]*2, '--r',␣
        ↪label='Median HR')
      plt.plot(t[[0, -1]], [60/np.mean(np.diff(t[fqrs_peaks]))]*2, ':k', label='Mean␣
        ↪HR')
      plt.legend()
      plt.xlabel('Time (s)')
      plt.ylabel('Fetal Heart Rate (bpm)')
      plt.title('Fetal Heart Rate Estimation')
      plt.show()
```
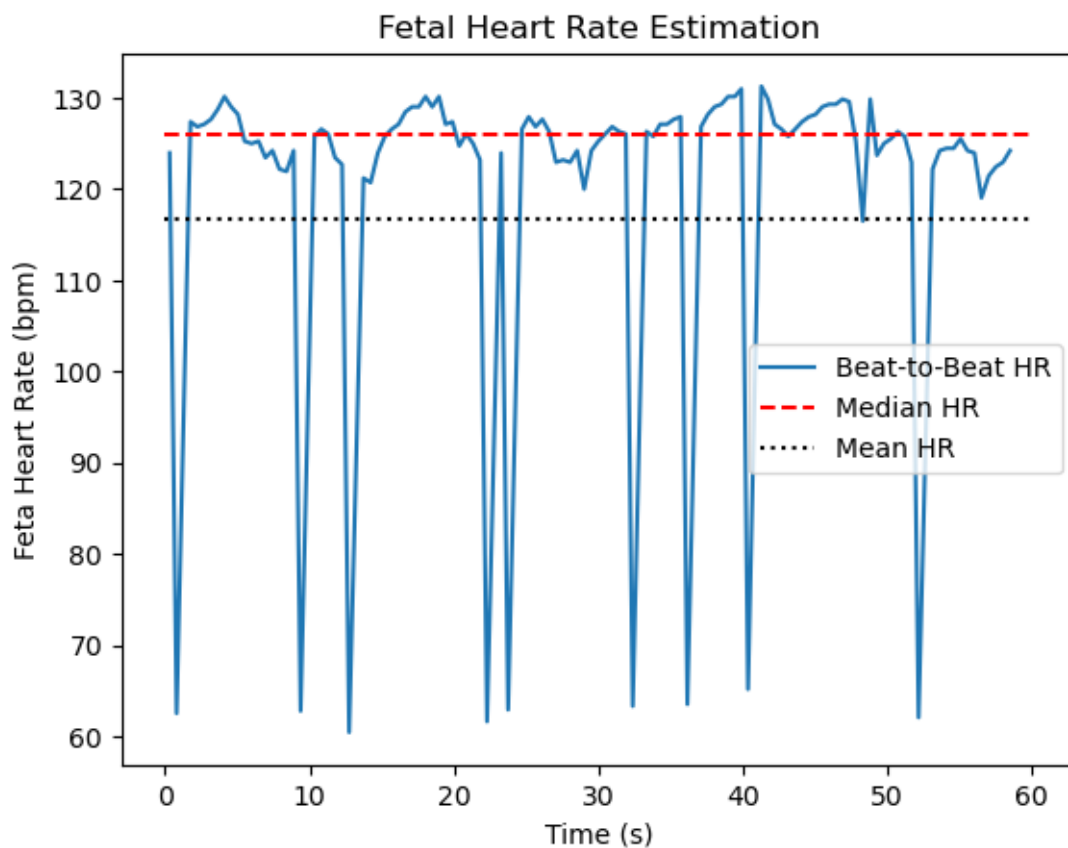


## 2.3   Question 3

Determine the follwing three values:

- i) the average amplitude of the maternal QRS peaks (mQRS);

- ii) the average amplitude of the fetal QRS peaks (fQRS);

- iii) the ratio between the average amplitudes of i) mQRS and ii) fQRS peaks.

```
[13]:  # amplitudes and their ratios
       # estimate mqrs amplitudes
       mqrs_amplitudes = maternal_ecg[mqrs_peaks]
       print('Maternal MQRS Amplitude: mean={:.1f}; median={:.1f}'.format(np.
        ↪mean(mqrs_amplitudes), np.median(mqrs_amplitudes)))
       # estimate fqrs amplitudes
       fqrs_amplitudes = fetal_ecg[fqrs_peaks]
       print('Fetal MQRS Amplitude: mean={:.1f}; median={:.1f}'.format(np.
        ↪mean(fqrs_amplitudes), np.median(fqrs_amplitudes)))
       # their ratio
       print('Ratio Fetal/Maternal QRS Amplitude: {:.3f}'.format(np.
        ↪median(fqrs_amplitudes) / np.median(mqrs_amplitudes)))
       print('Ratio Maternal/Fetal QRS Amplitude: {:2.2f}'.format(np.
        ↪median(mqrs_amplitudes) / np.median(fqrs_amplitudes)))
```

```
Maternal MQRS Amplitude: mean=103.2; median=102.1
Fetal MQRS Amplitude: mean=20.1; median=19.9
Ratio Fetal/Maternal QRS Amplitude: 0.195
Ratio Maternal/Fetal QRS Amplitude: 5.14
```

## 2.4  Question 4

How many of the principal components of the first PCA clearly show a maternal ECG signal?
Which ones?

The first two: `PC1[:, 0]` and `PC1[:, 1]`

## 2.5  Question 5

How many of the principal components of the second PCA clearly show a fetal ECG signal? Which
ones?

The first one: `PC2[:, 0]`

## 2.6  Question 6

Not all of the fetal QRS peaks seem to be detected properly. Do you have an explanation why
this happens and under which circumstances? Is it a problem of the fQRS detector, the mQRS
cancellation or of another block of the algorithm?

Mainly all fQRS peaks which do not get properly detected occur closely to mQRS peaks. This
indicates that the mQRS cacellation is not perfect, i.e., it is difficult to separate the mQRS of high
amplitude with the fQRS of much smaller amplitude but of very similar morphology. While some
missed fQRS are virtually not visible in the `fetal_ecg` signal, others are of smaller amplitude than
the surroundin fQRS peaks. For the latter the missing detections can thus also be partly attributed
to the QRS peak detector.