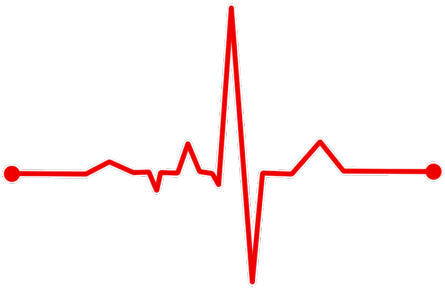

UL FRI Data Science - Biomedical signal and image processing
Project 3 Report

Analysis of electrocardiographic (ECG) signals - heartbeat classification

Žiga Trojer

Abstract - for the purpose of the third seminar paper in the subject of OBSS, we implemented the procedure of classification of heartbeats ECG signal. The algorithm works good as it uses different approaches such as correlation coefficient and automatic adjustment of threshold according to different ECG. When using classification, the evaluation of the algorithm on the MIT-BIH database [1] [2] gives 95.68 % average sensitivity, 94.55 % average positive predictivity and 53.26 % average specificity.

Introduction and overview.

QRS detection is a very difficult problem as there are many factors present that make detection difficult. These are, for example, the variability of the QRS complex and various noises that are present in the signal - muscle noises (occurs due to the movement of the electrodes). There is also a T - wave, which can be misidentified with a QRS complex. QRS detection provides an important basis for automated beat diagnosis.

In this paper, we will classify a heartbeat either as normal heartbeat or abnormal, ventricular heartbeat (also called premature ventricular contraction - PVC). It is a relatively common event where the heartbeat is initiated by Purkinje fibers in the ventricles rather than by the sinoatrial node. PVCs may cause no symptoms or may be perceived as a "skipped beat" or felt as palpitations in the chest. Single beat PVCs do not usually pose a danger [3].

ECG filtering.

The first step to automatically suggesting a diagnosis to a doctor is to successfully detect all QRS complexes. This can usually be quite a challenge, as we face the following problems:

- A baseline wander disorder is present in the electrocardiogram (low frequency interference).
- Noise due to electrode movement is present (high frequency interference).
- Noise due to poor contact between the skin and the electrode is present.

The useful frequency content of signal ranges from 0.6 Hz to 30 Hz. This means that we can solve two of the three listed problems with digital filters. With high-pass filters, we filter the signal by removing the baseline wander - placing the signals on a "common denominator". With low-pass filters, noise due to electrode movement can be removed, since the frequency of noise is above the 30 Hz. However, there is still a problem with noise, which is present due to poor contact of the electrodes with the skin, since it turns out that this noise has interference in the range of 0.6 to 30 Hz. If we use a filter here, we would also be filtering important information about the signal.

There are two types of filters. These are finite impulse response (FIR) and infinite impulse response (IIR) filters. FIR filter is a filter of finite duration, because it settles to zero in finite time. In general, FIR filter difference equation can be written as

$$y[n] = \sum_{i=0}^{N} b_i x[n-i],$$

where $x[n]$ is the input signal, $y[n]$ the output signal, N is the filter order and $b_i$is the value of the impulse response of the i-th instant. Impulsive response of the filter is the infinite sequence

$$h[n] = \begin{cases} b_n, & 0 \leq n \leq N \\ 0, & \text{otherwise.} \end{cases}$$

The FIR filter can also be written as convolution $y[n] = x[n] * h[n]$. A nice feature of the FIR filters we use in the analysis is that they have a linear phase. We will highlight the importance of this soon.

On the other hand, IIR filter is a filter of infinite duration, in which the impulse response does not become exactly zero

past a certain point. IIR difference equation is the following

$$y[n] = \sum_{m=0}^{M} b_m x[n-m] + \sum_{k=1}^{K} a_k y[n-k],$$

where $x[n]$ is the input signal, $y[n]$ the output signal and $max\{M,K\}$ is the order of the filter. We can also find the transfer function of the filter, by taking the Z-transform (most important property here is time-shift property), which converts a discrete-time signal into a complex frequency domain representation. Transfer function of IIR filter is in this form

$$H(z) = \frac{\sum_{m=0}^{M} b_m z^{-m}}{1 + \sum_{k=1}^{K} a_k z^{-k}}.$$

Knowing the transfer function allows us easy to determine whether or not a system is stable. That is, all poles must be located within a unit circle in the z-plane.

FIR or IIR filters can be used to filter the signals - each of these has advantages and disadvantages. The advantage of IIR filters is that they are simple, that is, the filter requires only a few coefficients. A small number of coefficients, however, means a small delay in the output signal. The disadvantage of IIR filters, however, is that the output signal is deformed - this is due to the nonlinear phase of the filter.

The FIR filter, on the other hand, has a linear phase - the output is not deformed. Thus, the signal retains those characteristics that are useful for QRS complex detection. Because we want to have filters with very sharp characteristics, such FIR filters have a lot of coefficients. A large number of coefficients, however, means a large delay in the output signal. This means that FIR filters cannot be used for real-time detection. However, they are very useful when processing signals later (as with Holter monitoring). [4]

Detecting QRS complexes.

The next step is the detection of QRS complexes - Figure 1 shows QRS detection pipeline. The first step is signal processing. This contains of two important parts. The first part is applying a linear filter by ”boosting ” the slopes and peaks of the QRS complex in a signal. Then we apply a nonlinear transformation - the signal is squared and a low-pass moving-average filter is applied. The result is a detection function, on which we then detect fiducial points. Those are the peaks of the detection function, so these can be identified by different threshold techniques.

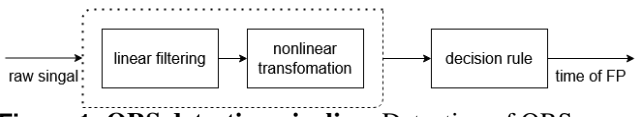

Figure 1. QRS detection pipeline. Detection of QRS complexes consists of 2 major parts. First one is processing the input signal to get detection function. Second part is to define a decision rule to detect at fiducial points of QRS complexes.

Here, too, various problems or challenges can arise. Irregular pulses may occur - pulses with small amplitude, the presence of high T-waves and P-waves, abnormal length of the QRS complex, etc.

In general, the detection function can be written as

$$d[n] = G[(\sum_{i=1}^{N}(|H_1(x_i[n])| + |H_2(x_i[n])|))^2],$$

where H_1 is a filter sensitive on slopes, H_2 filter sensitive on peaks and G low-pass moving average filter. Figure 2 shows an example of detection function. Positions of fiducial points are marked with a red vertical line.

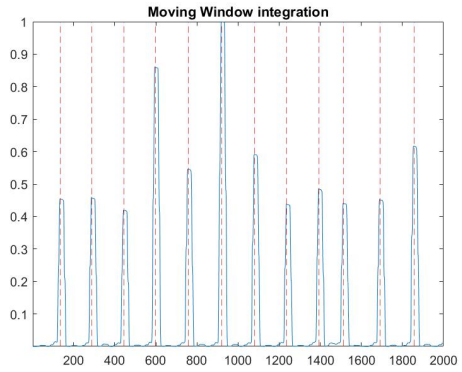

Figure 2. Detection function and detected fiducial points. Detection function of a recond ‘100‘ of MIT-BIH database [2]. Here, fiducial points were detected with adapting threshold, described in the Pan-Tompkins article [5].

Classification of heartbeats.

We are left with the last part, the classification of heartbeats. Figure 3 shows the heartbeat classification pipeline. When we have positions of the QRS complexes, we extract features from the neighborhood of the fiducial point and based on this feature vector, we make a decision whether a heartbeat is a normal or ventricular beat.

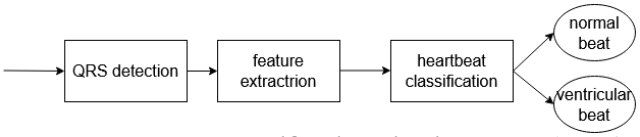

Figure 3. Heartbeat classification pipeline. It consists of three main parts. First part is the QRS detection, which we already covered. Second part is the feature extraction, which prepares some features, based on which an algorithm decides, what type of heartbeat it is.

For the feature extraction step, we first need to normalize the signal. With other words, we need to put all heartbeats on the same denominator - drift suppression. There are three possible ways to do this:

- Use of high-pass recursive filter for drift suppression.

- Estimation of isoelectric level.
- Automatically seeking for the position of the isoelectric level.

After applying one of the above techniques, our signal is ready for feature extraction. We would like to derive the morphologic feature vectors from the ECG pattern vector. In that way, we reduce the data dimensionality; the dimension of feature vector y is greatly less than the dimension of the pattern vector x. There are several feature extraction techniques, such as:

- Using norms of linear algebra or correlation to distinguish QRS complex morphologies.
- Using non-linear trimmed moving average filters.
- Using ortonormal function model transform-based feature extraction technique - Karhunen–Loève coefficients.

Let us define some of the norms that are used with the first approach: first norm $d_1 = \frac{1}{N}(|x_1 - y_1| + \ldots + |x_N - y_N|)$, second norm $d_2 = \sqrt{\frac{1}{N}(|x_1 - y_1|^2 + \ldots + |x_N - y_N|^2)}$, infinite-norm $max\{|x_1 - y_1|, \ldots, |x_N - y_N|\}$ and dissimilarity norm, which we will define later. We calculate these norms for points in some neighborhood of a fiducial point. It turns out that it is best to take an interval $[\mathrm{FP} - 60\mathrm{ms}, \mathrm{FP} + 100\mathrm{ms}]$. However, the question is, based on what do we calculate the norm? The answer is, it is based on the average normal heart beat estimated from the beginning of the signal. Norm tells us how similar observed heartbeat is to the average heartbeat - higher the norm, greater the dissimilarity. Norm of normal heartbeats will be lower compared to the norm of the ventricular heartbeat. Now we need to set a threshold, from which value on there is a ventricular heartbeat. In other words, we compare average normal heartbeat by every other heartbeat and we measure the (dis)-similarity between them.

Methods

We implemented the algorithm in Matlab. We also used Cygwin64 for data preparation purposes. We downloaded the data from the Physionet, where we used the MIT-BIH Arrhythmia Database [6] [2]. This one contains half-hour ECG signals from 47 subjects - we used 39 of them, as 8 records were causing me problems in preparation - for those records, there were no normal heartbeats in the first 5 minutes of the signal. Sampling frequency of those records was 360 $\frac{\text{samples}}{\text{second}}$. We then had to transform all the signals to the appropriate Matlab format. Below are parts of the code we used for transformation.

Listing 1. Code for transforming signals.

```
xform -i 100 -a atr
wfdb2mat -r 100
```

After a successful data transformation, we also calculated an average normal heartbeat for each of the record. We considered the first 5 minutes of the signal, where we took only normal heartbeats for the average heartbeat. We have identified positions of fiducial points from already given annotations. Neighborhood of $[\mathrm{FP} - 60\mathrm{ms}, \mathrm{FP} + 100\mathrm{ms}]$ was used. This means that we took 22 samples left from the FP and 36 samples right from the FP. Below is part of the code, used for calculating the average heartbeat.

Listing 2. Code for calculating average heartbeat.

```
sigavg -r 100 -a atr -f 0 -t 300 -p N -d -0.060
    0.100 >avg100.txt
```

Adjusting QRS complexes by amplitude.

We first had to adjust all the QRS complexes by amplitude. The reason is because drift is present in the signal. If we have not done this, the differences between the reference/average heartbeat and the normal heartbeat in the drifted signal would be so large that the algorithm would conclude that it is a ventricular heartbeat.

We used a high-pass recursive filter for drift suppression - see Figure 4. It is defined as

$$H(z) = \frac{c_1(1 - z^{-1})}{(1 - c_2 z^{-1})},$$

where $c_1 = \frac{1}{1 + tan(F_c \pi T)}$ and $c_2 = \frac{1 - tan(F_c \pi T)}{1 + tan(F_c \pi T)}$. This is IIR filter, since it has a difference equation

$$y[n] = c_2 y[n-1] + c_1(x[n] - x[n-1]).$$

Cut-off frequency of this filter will be at $F_c = 2.2Hz$ and sampling period will be $T = \frac{1}{360}$. Our filter is now

$$H(z) = \frac{0.9812(1 - z^{-1})}{(1 - 0.9623 z^{-1})}$$

From before we know that IIR filter has nonlinear phase - the output signal is a bit deformed. But for our classification problem it does not bother us too much, since the differences of used metric between normal and ventricular heartbeats will be big enough for distinguish between them.

Calculating the feature vector.

For calculating the difference between the currently considered heartbeat and the reference, we used the dissimilarity norm. It is defined as:

$$d_r = \begin{cases} 1 - r, & \text{if r} > 0 \\ 1, & \text{otherwise.} \end{cases}$$

where r is a measure of dissimilarity, defined as

$$r = (S_x S_y)^{-\frac{1}{2}} \sum_{i=1}^{N} ((x_i - x_{avg})(y_i - y_{avg})).$$

Here, y and x denote samples of reference and currently considered heartbeat. First two variables are defined as $S_x = \sum_{i=1}^{N}(x_i - x_{avg})^2$ and $S_y = \sum_{i=1}^{N}(y_i - y_{avg})^2$. This means that r is measuring the correlation between the reference and currently considered QRS complex. Higher the d_r norm, higher the difference.

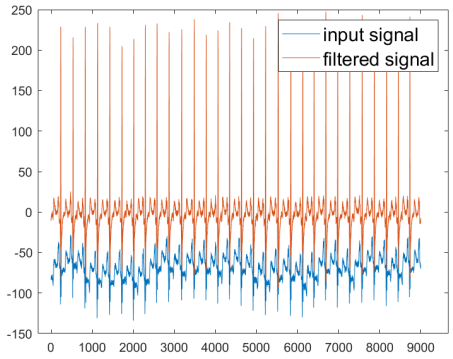

Figure 4. Signal before and after adjusting QRS complexes by amplitude. High-pass recursive filter for drift suppression was used. It is nicely seen that there is no more drift in the signal.

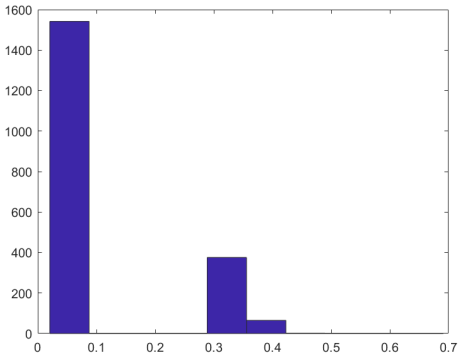

Figure 5. Distribution of dissimilarity norm of a record '119'. Threshold t_2 is determined by searching for the biggest gap between dissimilarity norms.

Setting the classification threshold.

For each heartbeat, we have calculated the dissimilarity norm. We need to set the classification norm to distinguish between normal heartbeats and ventricular heartbeats. The easiest way is to empirically determine the fixed threshold. This was done on the entire MIT-BIH database [1] [2] and the optimal threshold was 0.8. Heartbeats with dissimilarity norm above this value were classified as ventricular heartbeats. We also tried other norms, but we got the best results with the dissimilarity norm.

Because we wanted to improve our classification algorithm, we implemented our own adapting threshold. We first calculated two thresholds - t_1 and t_2. First, we ordered values of dissimilarity norms descending. Threshold t_1 is the value of dissimilarity norm, where the difference of two consecutive values is higher than one standard deviation of those norms. If there were more such candidates, we took the lower threshold.

To determine threshold t_2, we searched for the "biggest gap" of norms. Figure 5 shows distribution of dissimilarity norms for one of the records of MIT-BIH database. We see that the biggest gap is between the first bin and the second bin. Our threshold is now set as the value of the right value of the gap: $t_2 = 0.3$. Finally, we calculate our threshold, which we use in classification: threshold $= min\{t_1, t_2\}$.

Results

We used the MIT-BIH database [1] [2] for evaluation. The table 1 contains the results of the improved algorithm - the algorithm was tested on 39 records, half an hour long. We used 3 metrics: sensitivity, which is defined as Se $= \frac{\mathrm{TP}}{\mathrm{TP+FN}}$, positive predictivity $+\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP+FP}}$ and specificity Se $= \frac{\mathrm{TN}}{\mathrm{TN+FP}}$. The average sensitivity is 95.6%, the average positive predictivity is 94,5% and average specificity is 53.2%. There are few records where there are no ventricular heartbeats - this means that we can not calculate the specificity for those records. Our algorithm only works with records, where there is at least one normal heartbeat in the first 5 minutes of the signal.

It turns out that we quite a lot improved the algorithm with the adapting thresholds. Evaluating algorithm with the fixed threshold of 0.8 on the same records, we get an average sensitivity of 98,5%, average positive predictivity 94,0% and average specificity 31,0%. Although the sensitivity is lower with adaptive thresholds, positive predictivity and specificity are higher, which is an important fact. In our case, the sensitivity is the proportion of the normal heartbeats that were correctly classified as normal. On the other hand, specificity tells us the proportion of ventricular heartbeats that were correctly classified as ventricular heartbeats. Positive predictivity tells us the proportion of classified normal heartbeats which were actually normal heartbeats. For the purpose of heart rate classification, we find the specificity metric the most important, because our purpose is precisely to suggest a diagnosis of whether ventricular heartbeats are present. Therefore, we will be happy to compromise and increase the other two metrics by lowering the sensitivity. The most important thing is to correctly classify as many ventricular heartbeats as possible, which tells us exactly the specificity metric.

Discussion

The algorithm is very well designed and works just fine given its simplicity. However, it is also necessary to understand here that the algorithm distinguishes only between ventricular and normal heartbeats. Other heartbeats were removed from the analysis. Of course, our algorithm would not be useful for clinical purposes, as we do not know how the algorithm would classify other types of heartbeats. It is also necessary to be aware that we already knew the location of the fiducial point in advance for the purpose of this task, which is not possible in practice. For a good classifier, we need to have a very good detector. It would be quite a challenge to implement such algorithm, that would classify heartbeats in real time - a lot of challenges here, as for example, we would not know the reference heartbeat and we would need some other technique.

Table 1. The performance of the algorithm.

Record	TP	FP	FN	TN	**Se**	**+P**	**Sp**
100	1870	0	1	1	0,999	1	1
101	1516	0	1	0	0,999	1	-
103	1723	0	3	0	0,998	1	-
104	97	0	0	1	1	1	1
105	2112	29	8	0	0,996	0,986	0
106	1234	410	1	50	0,999	0,750	0,108
108	1264	2	195	11	0,866	0,998	0,846
112	2106	0	2	0	0,999	1	-
113	1498	0	2	0	0,998	1	-
114	0	0	1558	4	0	1	-
115	1633	0	3	0	0,998	1	-
116	1900	97	17	1	0,991	0,951	0,010
117	665	0	617	0	0,518	1	-
119	1295	0	1	364	0,999	1	1
121	1555	0	2	1	0,998	1	1
122	2051	0	2	0	0,999	1	-
123	1264	0	1	3	0,999	1	1
200	1437	90	0	610	1	0,941	0,871
201	1184	196	0	2	1	0,857	0,010
202	1799	12	0	3	1	0,993	0,2
203	2042	45	60	328	0,971	0,978	0,879
205	2121	0	1	65	0,999	1	1
208	1303	819	4	5	0,996	0,614	0,006
209	2145	0	0	1	1	1	1
210	2009	33	0	131	1	0,983	0,798
212	790	0	3	0	0,996	1	-
213	2208	160	0	35	1	0,932	0,179
215	2660	130	0	1	1	0,953	0,007
217	235	135	0	4	1	0,635	0,028
219	1714	49	0	2	1	0,972	0,039
220	1599	0	1	0	0,999	1	-
221	1703	307	0	9	1	0,847	0,028
222	1687	0	7	0	0,995	1	-
223	1655	373	0	82	1	0,816	0,180
228	1396	282	1	20	0,999	0,831	0,066
230	1857	0	0	1	1	1	1
231	300	0	1	0	0,996	1	-
233	1858	240	0	452	1	0,885	0,653
234	2237	0	0	3	1	1	1
39 records	59722	3409	2492	2216	0,959	0,946	0,393
				average:	0,956	0,945	0,532

References

[1] Goldberger A., Amaral L., Glass L., Hausdorff J., Ivanov P. C., Mark R., and Stanley H. E. Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals. circulation. *Circulation [Online]*, 101(23):e215–e220, 2000.

[2] Mit-bih arrhythmia database. https://physionet.org/content/mitdb/1.0.0/. Accessed: 2021-04-01.

[3] E. P. Gerstenfeld and T. De Marco. Premature ventricular constractions. *Circulation*, 140(8):624–626, 2019.

[4] V. Oppenheim Alan, S. Willsky Alan, and T. Young Ian. In *Signals and Systems*, page 256, 1983.

[5] J. Pan and W. J. Tompkins. A real-time qrs detection algorithm. *IEEE Transactions on Biomedical Engineering*, BME-32(3):230–236, 1985.

[6] Moody GB and Mark RG. The impact of the mit-bih arrhythmia database. *IEEE*, Eng in Med and Biol 20(3):45–50, (May - June 2001).