

QRS Complex Classification

Assignment 2.b

Biomedical Signal and Image Processing 2020/21, Faculty of Computer and Information Science, University of Ljubljana

Urša Zrimšek

Abstract—This is a report of the second assignment for the course Biomedical Signal and Image Processing. It follows the methods proposed in [1], with some minor improvements. The algorithm is evaluated on data from [2] on which it reaches sensitivity of 94.26%, positive predictivity of 98.5% and specificity of 85.1%.

I. INTRODUCTION

QRS complex classification is the next step after QRS detection. In our algorithm we only did classification, and took the indices of the detected heartbeats as input. We were classified only normal and ventricular heartbeats, so first we prepared the reference files from database [2] to only include N and V. For good classification we need to filter out the noise and emphasize the biggest changes in the measured voltage, to be able to then find the height of the main spike on the ECG, that should be much higher for normal heartbeats than for ventricular. The algorithm doing that is described in the next section. After that we will analyze the results and propose further improvements in the discussion.

II. METHODOLOGY

This detection system is composed of three stages: *nonlinear high-pass filtering*, *nonlinear low-pass filtering* and of *pattern classification*.

A. Nonlinear high-pass filtering stage

This stage is a combination of a *trimmed moving average filter* and *delayed system* with delay of $\frac{M+1}{2}$. If x are the values in the ECG record, we can characterize TMAF as y_1 and the delayed system as y_2

$$y_1[n] = f^t T\{x_n\}, y_2[n] = x[n - \frac{M+1}{2}],$$

where $T\{x_n\}$ is a vector of last M values of x ranked from smallest to largest, and

$$f_k = \begin{cases} 0; & \text{for } k = 1, \dots, N \text{ or } k = M - N + 1, \dots, M \\ \frac{1}{M - 2N}; & \text{otherwise} \end{cases} \quad (1)$$

That means we remove N largest and N smallest values and average others. If $N = 0$ the TMAF is actually a moving averaging process (y_1 is equal to mean of last M values of x), and if $N = \frac{M-1}{2}$ it is called the moving median process (y_1 is equal to median of last M values of x). We then combine them as

$$y_b[n] = y_2[n] - y_1[n].$$

We can see the graphical result on Figure 1.

B. Non-linear low-pass filtering stage

In this stage the output of HPF is squared point-by-point, then we go over it with moving window summation. We can see the output of LPF on Figure 2.

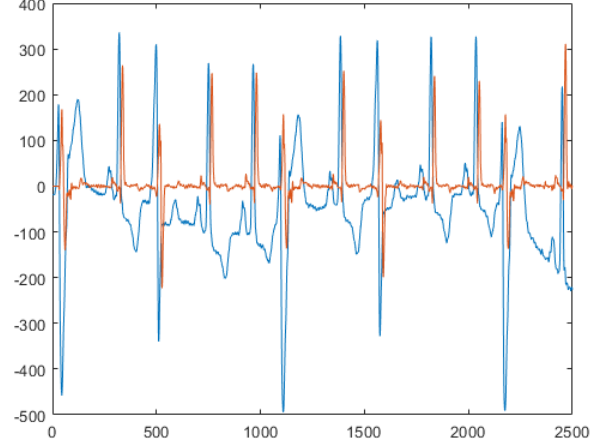


Figure 1. Original signal (blue) from record 233 and output of high-pass filtering stage (orange).

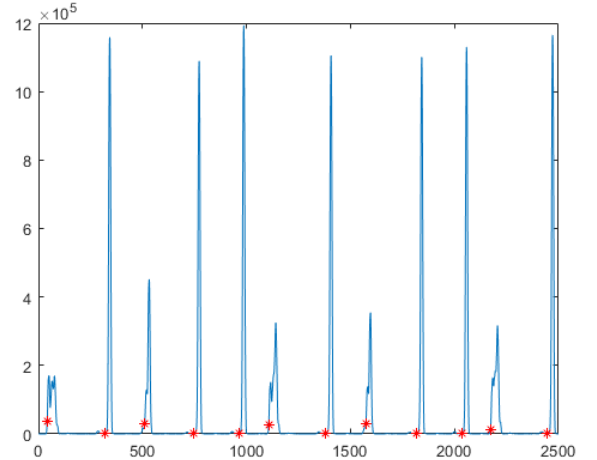


Figure 2. Output of non-linear low-pass filtering stage of record 233. We can see that the ventricular heartbeats (1st, 3rd, 6th, 8th and 10th) have much lower peak then the normal ones. Red * mark the previously detected QRS complexes.

C. Pattern classification

In [1] there is an adaptive threshold, that updates itself at every normal QRS complex by the rule:

$$threshold = \beta \times 0.4 \times PEAK + (1 - \beta) \times threshold.$$

We can interpret β as the forgetting factor that weights the importance of new peak against the previously set threshold.

At the beginning we set the threshold to the lowest value of all the peaks of normal beats. We took the information from reference files, which would in real life situation mean that at

the beginning the doctor was present, and he specified which beats are normal and which are ventricular.

We then take the maximum value (*PEAK*) of the NLPF output on a small neighbourhood of every ground truth index that is given as an input to our algorithm. We then classify it as a normal heartbeat if the peak exceeds the threshold, and as ventricular otherwise.

D. Improvements

The first improvement was to do the above steps on both leads, add them together and apply the decision making on the sum.

With the algorithm described in [1], we can reach high sensitivity (if we take normal heartbeats for positives, and ventricular for negatives), but specificity stays low. The change that made possible for specificity to be better was to set the 0.4 constant from the threshold adaptation formula to higher values.

III. RESULTS

We were evaluating the results based on *sensitivity*, *specificity* and *positive predictivity* on data from [2]. The metrics are defined as:

$$Sensitivity = \frac{TP}{TP + FN}$$

$$Specificity = \frac{TN}{TN + FP}$$

$$Positive\ predictivity = \frac{TP}{TP + FP}$$

To get the best results there were still some parameters we needed to adjust. In the NHPF stage we needed to choose value of M and N . The study in [1] suggests we use $M, N = (7, 5)$, but we got better results with bigger values. Specificity was at most 23% with all the other improvements, when using this two values, and it reached almost 93% with $M = 31$ and $N = 15$. This were the values where we got the best results, and we can see, that this is in fact moving median process, not trimmed average. Here we can also mention that in our testing higher N brought us higher specificity and a bit lower sensitivity.

In LPF we needed to set the window size for the summation - K . We tried $K \in \{10, 20, 30, 40\}$ and got best results with $K = 10$.

In decision making stage, we needed to set β and we also found out that specificity can improve a lot by changing the 0.4 constant in the threshold adaptation. We tried different values for forgetting factor: $\beta \in \{0.01, 0.05, 0.1, 0.15, 0.5\}$. The best results were obtained with $\beta = 0.05$.

For the constant, we can't say which results are better, since there is much smaller specificity when we maximize sensitivity and vice-versa. Bellow you can see the results with different constants at selected values: $M = 31, N = 15, K = 10, \beta = 0.05$.

Constant	Se	Sp	+P
0.4	99.01	67.51	96.49
0.5	96.01	80.60	98.09
0.6	94.26	85.10	98.50
0.8	75.61	91.14	98.88
0.6 *	94.07	78.27	97.82

The above optimizations of parameters were already made on the improved algorithm - summed leads. If we would do the filtering only on one lead the results would be as shown in the last line of the table (*).

IV. DISCUSSION

For the final (best) algorithm we chose the one where constant equals 0.6, because it maximizes the metric with the lowest value, but you should choose the parameters based on which metric you want to maximize. If the most important thing is not to miss any ventricular heartbeats, you should maximize specificity and choose bigger constant.

If we check the results for each record, we can see that specificity is either really good or really bad. For further improvements we should try to improve the classification on those records where specificity is 0, or even bellow 10%. Usually these are records that have a low number of ventricular beats.

Further problem with the algorithm could be, that if we come across very high peak (could be an abnormality at measuring) and the threshold becomes higher than any normal heartbeat, it won't update itself anymore, and all the further heartbeats will be detected as ventricular. We should find a solution for this problem.

REFERENCES

- [1] S. Chen and H. Chen, "Development of a real-time qrs beat classifier using a nonlinear trimmed moving averaging-based system," in *Computers in Cardiology, 2003*. IEEE, 2003, pp. 577–580.
- [2] G. B. Moody and R. G. Mark, "The impact of the mit-bih arrhythmia database," *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, no. 3, pp. 45–50, 2001.