Detection of Heart Beats in ECG Signals

Assignment #1

Biomedical Signal and Image Processing 2019/20, Faculty of Computer and Information Science, University of Ljubljana

Jernej Vivod

***Abstract*—The first assignment for the Biomedical Signal and Image Processing class required students to implement a heart beat detection algorithm to detect heart beats from ECG signals. The detection process could also be augmented with auxiliary physiological data that is available with some signals contained in the databases used to evaluate the implemented algorithm. For this assignment, I implemented the algorithm outlined in the paper Robust Detection of Heart Beats using Dynamic Thresholds and Moving Windows by Marcus Vollmer [1]. The implementation mostly follows the procedure described in the paper but takes certain liberties and makes some extensions. The results of evaluations show that the algorithm performs very well especially for noisy signals and data, where the ECG signal is obfuscated.**

I. Introduction

Heart beat detection in ECG signals is an important research topic motivated by the ever increasing need for automated and large-scale physiological data analysis for which many algorithms have been developed. This report describes the implementation and evaluation of the algorithm outlined by Vollmer [1]. The algorithm is designed to be especially resilient to noise and selects relevant channels in multivariate recordings using a probabilistic approach.

II. The Algorithm

The signal is first filtered with a trimmed moving average filter that acts as a high-pass filter. The α%-trimmed moving average filter with window length ω of a time series $(x_t)_{t=1,\ldots,n}$ is defined as

$$TMA_i := \frac{1}{\omega - 2k} \sum_{j=k+1} \omega - k\tilde{x}_j$$

with $k = \lceil \frac{\omega\alpha}{2} \rceil$ and sorted values $\tilde{x}$ of $(x_t), t \in [i - \frac{\omega}{2}, i + \frac{\omega}{2}$.

The parameters for this filter proposed by Vollmer performed poorly and a much smaller value of ω and α were used for the final evaluation.

The signal was then standardized and filtered with a range filer. The range is the difference between local maximum and local minimum in the neighborhood of a particular time point. This filter emphasises quick transitions in the signal and transforms it in a form that is particularly suitable for heart beat detection.

The range signal was then used to extract the beat positions. To dinamically compute the threshold, local maxima and minima of the signal were interpolated using cubic splines. The local extrema were filtered with a median filter to prevent fitting the splines to outliers.

The dynamic threshold for position i was computed as

$$r_i > \mathrm{INT}_{max_i} - 0.3 \cdot |\mathrm{INT}_{max_i} - \mathrm{INT}_{min_i}|$$

This formulation of the threshold ensures that the threshold always lies between the interpolated local maxima and local minima and is different than the simpler author's proposed

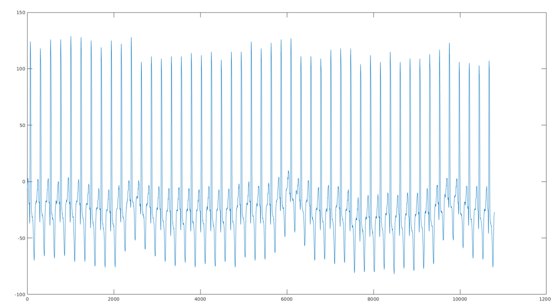

Figure 1. Subset of a typical ECG signal.

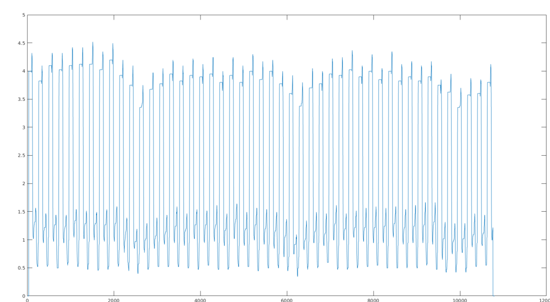

Figure 2. Signal after range filtering

formula. The r_i value was also required to match the next $F_s/25$ samples of the range signal as a consistency criterion.

The algorithm augments its detecting capabilities by dynamically selecting relevant channels in multivariate records. This is done by probing each channel using a 30 second signal subset. A channel is considered "safe" if the probability that the relative RR interval lies on the interval $[0.8, 1.2]$ is greater than 0.8. The RR interval is the time between QRS complexes. The relative RR interval is defined as

$$\mathrm{relRR}_i := \frac{RR_{i+1}}{RR_i} \mathrm{for} i = 1, \ldots, n-1$$

This measure serves as a proxy to evaluate the amount of noise in a channel.

The "safe" channels were then cross-checked and beats were only declared as detected if there was a detection in each channel after the delays have been rectified. The rectification of delays was computed dynamically if there was an ECG channel present in the subset of "safe" channels. If not, a static delay was applied.

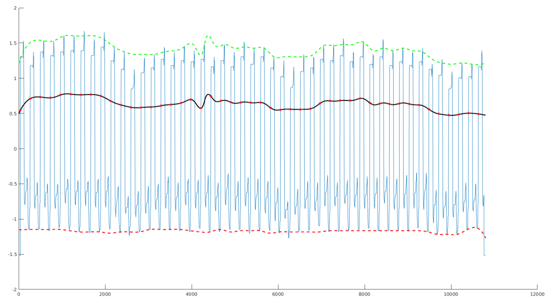

Figure 3. Signal with dynamic threshold for heart beat detection.

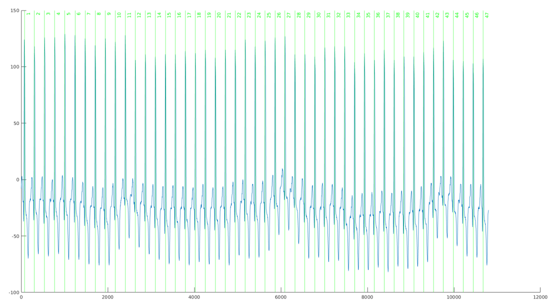

Figure 4. Beat detections plotted on original signal.

III. Methodology

The algorithm was implemented using the MATLAB programming environment. The evaluation process was automated and works by evaluating all mat files in a specified folder. The result is a text file containing the statistics of the evaluation. The wfdb MATLAB toolbox was used to access the channel descriptions. The script *detect_qrs* can be used to plot the detection process by setting the *plt* parameter to true.

IV. Results

The algorithm was evaluated on a sample of records from freely available databases. The statistics of the evaluations are included in the source code directory as text files with the prefix *results*.

V. Conclusion

The algorithm outlined in the paper presents a robust method of detecting heart beats in multivariate records and performs well even in the presence of noise. Due to assumptions and some simplifications, the implementation did not match the author's claimed results. Still, the code presents a good framework and proof of concept for further research. The algorithm also showed large sensitivity to its parameter values which were set empirically by trial and error. A systematic approach to tune the parameters could improve the success rate by a notable amount.

References

[1] M. Vollmer, "Robust detection of heart beats using dynamic thresholds and moving windows," 09 2014.