Noninvasive Fetal QRS Detection Using Echo State Network

Mantas Lukoševičius, Vaidotas Marozas

Kaunas University of Technology, Kaunas, Lithuania

Abstract

The proposed method combines established cardiology-specific techniques based more on domain knowledge with powerful supervised general-purpose machine learning approaches that are more data-driven. After filtering and normalization, maternal QRS complexes are detected and averaged maternal ECG is removed. The key task of detecting fetal QRS complexes is performed by an Echo State recurrent neural Network (ESN) trained by supervised machine learning. The training of the model is made possible by the availability of correctly annotated training data. Finally, fetal QRS annotations are obtained by a statistics-based dynamic programming approach interpreting the outputs of the ESN. The proposed approach is quite generic and can be extended to other type of signals and annotations.

1.   Introduction

Monitoring of fetal ECG (fECG) and its parameters would provide important information about the fetal heart status and various distress factors. The problem, however, is difficult. Noninvasive fECG has low signal-to-noise ratio, is contaminated by the strong interferences: maternal ECG (mECG), fetal brain activity, myographic signals, movement artifacts. Forty years of research provided little to clinically significant advances in prenatal fetal ECG monitoring [1]. PhysioNet portal with its Challenge'2013 [2] took initiative to inspire researchers to turn to this old problem again with new methods and tools and try to move the field forward.

We propose here a new approach for fetal QRS (fQRS) detection and heart rate estimation based on supervised machine learning. It employs two innovative key components: (i) an "echo state" recurrent artificial neural network is trained to recognize fQRS, and (ii) several options of dynamic programming approaches are used to fuse information coming from sensors with estimated statistics of fQRS to find the most likely sequence of fQRS timings.

2.   Dataset

PhysioNet [2] provided a collection of one-minute fQRS recordings. Each recording contains four noninvasive abdominal leads. Though sampling frequency is the same 1000 Hz for all recordings, the instrumentation varied and had differing frequency response, resolution, and configuration. The data have been divided into three datasets. Dataset A (75 records) was a training set which included noninvasive fECG signals, as well as reference annotations marking the locations of each fQRS complex (by marking the time of R events). Dataset B (100 records) included noninvasive fECG signals only and was used for evaluation of the Challenge entries by the organizers. The last Dataset C was reserved for evaluation of open-source Challenge entries and remained secret. The Challenge was to produce a set of annotations (fQRS complex locations) that matches the non-disclosed references as nearly as possible. fQRS complex locations are annotated by marking the R events.

3.   Method

Figure 1 outlines the components and signal flow of our approach. Apart from repairing, filtering, and normalizing the data, the solution consists of two major steps: canceling the mECG from the signals and detecting fetal R events in the remaining signal. We will discuss these major blocks in more detail in the three upcoming subsections, elaborating more on the latter.

3.1.   Initial preprocessing

We first fix the invalid values (-32768) in the data that were not correctly converted to the digital form by ADC. We notice in Dataset A that these values fall bellow the dynamic range of the ADC, thus the invalid values were replaced with the valid minimum of the signal.

To eliminate the big and sharp jump in the signal values at the very beginning, which is observed in some Dataset B signals, we simply flatten the first 5 ms of every signal by assigning the end value at 5 ms for the entire interval. We then filter the signal using a bandpass filter leaving only
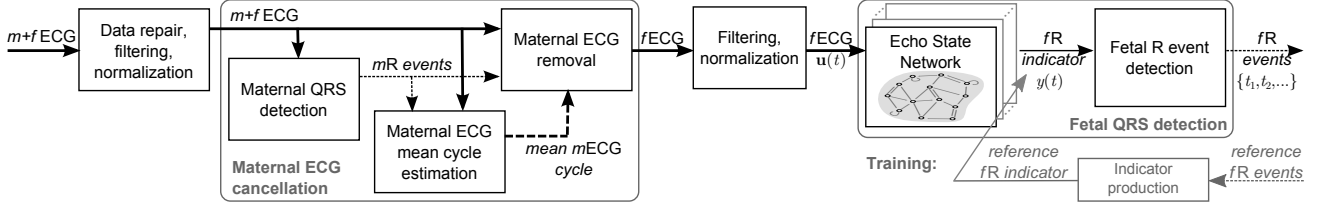
Figure 1. The block diagram of our approach. Here the bold arrows denote the four-lead ECG signals, and dotted lines denote signals that are not present for every time step.

the frequencies between approximately 5 and 48 Hz. The resulting signal is then normalized to have zero mean and unit standard deviation. All the preprocessing discussed above is applied to every of the four leads of the ECG signal where both maternal and fetal ECGs are present.

3.2. Maternal ECG removal

The *m*ECG removal is inspired by [3]. The implementation provided by the Challenge organizers was taken and improved upon in several respects. As depicted in Figure 1, the *m*ECG removal in turn consists of several steps.

First the *m*ECG complexes are detected on one of the four leads. The lead where the *m*ECG is most pronounced should best be used. We have observed that ECG has a characteristic asymmetric signal value distribution. We measure distribution asymmetry of each lead by computing *skewness* of it. The lead with maximal absolute skewness is selected for maternal QRS detection. In the process, all the leads are made to have a positive skewness, flipping the sign of the signal if necessary. This is a heuristic with a goal to make all *m*ECG R peaks point upwards. The algorithm of detecting maternal R peaks is mentioned in Section 3.3.

The mean *m*ECG cycle is computed by aligning all the *m*ECG cycles by the detected R points and averaging them in all the four leads separately. The mean *m*ECG is then subtracted around every R event from the original signal. The subtraction is done following [3] and taking special care of both ends of the signals.

The remaining signal is filtered again with a bandpass filter leaving only the frequencies between 12 and 48 Hz.

3.3. Fetal QRS detection

We approach the fetal QRS, or more precisely fetal R (*f*R), detection in the preprocessed signal as a supervised machine learning task. The procedure consists of two stages. The first stage is an artificial recurrent neural network of the type Echo State Network (ESN) [4–6]. It is trained in a supervised fashion to produce indications of *f*R events, using the annotations of the Dataset A to produce the target signals for the supervised training. The second stage takes the continuous *f*R indicator signals from the

trained ESN and interprets them in a probabilistic fashion to produce the discrete *f*R annotation events, making use of the statistics observed in the *f*R annotations provided in Dataset A. We discuss and test several alternatives for the second stage.

3.3.1. Echo State Network

We use an Echo State Network (ESN) [4–6] which is an artificial recurrent neural network with an update equation

$$\tilde{\mathbf{x}}(t) = \tanh\left(\mathbf{W}^{\text{in}}[1; \mathbf{u}(t)] + \mathbf{W}\mathbf{x}(t-1)\right), \quad (1)$$
$$\mathbf{x}(t) = (1-\alpha)\mathbf{x}(t-1) + \alpha\tilde{\mathbf{x}}(t), \quad (2)$$

where $\mathbf{x}(t) \in \mathbb{R}^{N_{\text{x}}}$ is a vector of ESN "reservoir" neuron activations and $\tilde{\mathbf{x}}(t) \in \mathbb{R}^{N_{\text{x}}}$ is its update, $\mathbf{u}(t) \in \mathbb{R}^{N_{\text{u}}}$ is the input signal, all at discrete time t, sampled at 1 kHz, $\tanh(\cdot)$ is applied element-wise, $[\cdot; \cdot]$ stands for a vertical vector concatenation, $\mathbf{W}^{\text{in}} \in \mathbb{R}^{N_{\text{x}} \times (1+N_{\text{u}})}$ and $\mathbf{W} \in \mathbb{R}^{N_{\text{x}} \times N_{\text{x}}}$ are the input and recurrent weight matrices respectively. In this task $\mathbf{u}(t)$ is the preprocessed and *m*ECG-removed signal with $N_{\text{x}} = 4$, and t is running from 1 to $T = 60\,000$ ms.

The readout from the the ESN is

$$\mathbf{y}(t) = \mathbf{W}^{\text{out}}[1; \mathbf{u}(t); \mathbf{x}(t)], \quad (3)$$

where $\mathbf{y}(t) \in \mathbb{R}^{N_{\text{y}}}$ is the network output, $\mathbf{W}^{\text{out}} \in \mathbb{R}^{N_{\text{y}} \times (1+N_{\text{u}}+N_{\text{x}})}$ the output weight matrix, and $[\cdot; \cdot; \cdot]$ again stands for a vertical vector concatenation. In this task $\mathbf{y}(t) = y(t)$ is the *f*R indicator signal with $N_{\text{y}} = 1$.

The weight matrices $\mathbf{W}^{\text{in}}$ and $\mathbf{W}$ are generated randomly according to some simple rules and parameters [7]. $\mathbf{W}^{\text{out}}$ is trained using linear regression [7] to produce the *f*R indicator signal $y(t)$. The target signal for the training was a zero signal spiking to one at *f*R events marked by the Dataset A annotations.

3.3.2. Statistical interpretation of the indicator signals

The last component in our architecture is responsible for interpreting the *f*R indicator signal $y(t)$ to produce discrete *f*R annotations $\{t_1, t_2, \ldots\}$. We interpret the signal $y(t)$ as an indication of the probability $P(t|\mathbf{u})$ that there is an *f*R

event at a time step t, given data $\mathbf{u}$. We must keep in mind, however, that it is not exactly trained to be the probability and especially the scaling of $y(t)$ can be way off, thus it should only be used as a comparative, but not an absolute value. Note also, that a probability of an event at a certain time step t is related to a probability density function over time by a constant dt which in our case is 1 ms.

Direct interpretation. The most straightforward approach of annotation is setting a threshold y_{th} and annotating every t_i with an fR event where $y(t_i) > y_{\text{th}}$. This would work but only for very clean signals, and setting of y_{th} would be problematic.

As in speech recognition, we can employ our knowledge of higher level statistics to improve the interpretation of information coming from lower level noisy signals. In speech recognition this would be context-specific dictionaries, probabilistic grammars, etc. In our context this would be the available statistics of fR events.

Employing R-R interval statistics. The direct method treats fR events at t_{i-1} and t_i as statistically independent. However, we know that fetal R-R (fRR) interval durations $(t_i - t_{i-1})$ typically lie in certain ranges. A more intelligent and widely popular approach is to set a permissible interval τ for the R-R values and find the next annotation at maximum inside the permissible interval:

$$t_i = \underset{t \in (t_{i-1}+\tau)}{\arg\max} \; y(t). \qquad (4)$$

This method worked better. This is in fact how our mQRS detection mentioned in Section 3.2 was implemented, except detecting maximum on one of the leads of the the raw signal $\mathbf{u}(n)$ instead of the trained indicator $y(t)$.

We can, however, make a better use of the fRR statistics. We compute a histogram of fRR durations from the Dataset A annotations and, after removing some outliers, we model their distribution as a Gaussian $\mathcal{N}\left(\mu_{f\text{RR}}, \sigma_{f\text{RR}}^2\right)$ with estimated parameters $\mu_{f\text{RR}} = 424\,\text{ms}$ and $\sigma_{f\text{RR}} = 46\,\text{ms}$. We thus estimate the probability of fRR duration being $(t_i - t_{i-1})$, instead of setting a hard interval, which is also the probability that fR occurs at t_i, given that it occurred at t_{i-1}: $P(f\text{RR} = t_i - t_{i-1}) = P(t_i|t_{i-1})$.

We combine the $P(t_i|\mathbf{u})$ (represented by $y(t)$) and $P(t_i|t_{i-1})$ to refine the fR annotations in two different algorithms.

In one we select t_i as

$$t_i = \underset{t \in (t_{i-1}+\tau)}{\arg\max} \; \big(y(t)P(t|t_{-1})\big) \qquad (5)$$

with the interval $\tau = (0, 2\mu_{f\text{RR}}]$.

In another variation we estimate maximal probability $p_{\max}(t)$ of fR for every time step t by dynamic programming

$$p_{\max}(t) = y(t) \underset{t_{-1} \in (t-\tau)}{\max} \big(p_{\max}(t_{-1})P(t|t_{-1})\big), \quad (6)$$

going forward in time and saving the argument $\arg\max$ t_{-1} for every t. Then we trace back the best $p_{\max}(t_i)$ from the end of the signal to reconstruct the sequence of annotations $\{\ldots, t_{i-1}, t_i\}$ leading to it. The interval τ in (6) is set around $\mu_{f\text{RR}}$, where $P(t|t_{-1})$ has high enough values. This is just sketching the idea of the algorithm. We actually substituted geometric mean for multiplication inside $\max$ of (6) to avoid shrinkage of $p_{\max}(t)$ in the process and thus disadvantaging short fRR durations by the mere fact that most probabilities involved are much smaller than 1 and $y(t)$ is not scaled as a probability.

Employing R-R interval variation statistics. The methods above treated durations of neighboring fRR intervals as statistically independent. However, we know that this is not the case. We compute a histogram of changes $f\Delta\text{RR}$ between durations of all two subsequent fRR intervals, and see that (excluding several outliers) it roughly follows a Gaussian distribution $\mathcal{N}\left(0, \sigma_{f\Delta\text{RR}}^2\right)$ with estimated $\sigma_{f\Delta\text{RR}} = 19\,\text{ms}$. This is a much narrower distribution than that of fRR. Thus modeled probability that a change in fRR duration will be $((t_i - t_{i-1}) - (t_{i-1} - t_{i-2})) = (t_i - 2t_{i-1} + t_{i-2})$ also corresponds to probability of fR at t_i, given fR at t_{i-1} and t_{i-2}: $P(f\Delta\text{RR} = t_i - 2t_{i-1} + t_{i-2}) = P(t_i|t_{i-1}, t_{i-2})$.

We created an algorithm based on that observation, similar to (6), which estimates a maximal joint probability of fR at t and last fRR interval having duration $t - t_{-1}$:

$$p_{\max}(t, t - t_{-1}) = y(t) \underset{t_{-2} \in (2t_{-1} - t - \tau)}{\max} \big($$
$$p_{\max}(t_{-1}, t_{-1} - t_{-2})P(t|t_{-1}, t_{-2})\big). \qquad (7)$$

This is again a sketch of a dynamic programming algorithm. After computing all $p_{\max}(t, t - t_{-1})$ in the forward direction and saving $\arg\max t_{-2}$ for every pair $(t, t - t_{-1})$, we take $t_i = \arg\max_t \max_{\Delta t} p_{\max}(t, \Delta t)$ at the end of the signal and trace back the sequence $\{\ldots, t_{i-1}, t_i\}$ leading to it, to get the fR annotations. The interval τ in (7) is centered around 0 where $P(t|t_{-1}, t_{-2})$ has high enough values.

fRR durations in algorithm (7) can take unlikely values (subject to admissible $(t - t_{-1})$ interval) with ambiguous signals, because there is no bias towards more probable fRR durations like in (6), but only towards probable changes in fRR durations. To introduce it, we can combine the two methods and have

$$p_{\max}(t, t - t_{-1}) = y(t) \underset{t_{-2} \in (2t_{-1} - t - \tau)}{\max} \big($$
$$p_{\max}(t_{-1}, t_{-1} - t_{-2})P(t|t_{-1})P(t|t_{-1}, t_{-2})\big) \quad (8)$$

instead of (7). We in fact used a geometric moving average instead of the multiplication inside the $\max$ in (7) and (8).

3.3.3. Parameter tuning and validation

We used 15-fold cross-validation on the 75 Dataset A annotated signals with an error function similar to Events 2 and 5 of the Challenge to test the many design options and parameters in all the components of our solution. ESNs enable us to do this massive cross-validation with minimal overhead [7]. Our solution was implemented in Matlab.

We have tried different parameter settings with our ESN networks (2)(3) following practices described in [7]. We used ESN reservoirs of size $N_x = 1000$, leaking rate $\alpha = 0.9$, spectral radius of the reservoir connections $\rho(\mathbf{W}) = 0.9$ or 0.94, $\mathbf{W}^{in}$ scaling of 0.1 or 0.08. In some of our solutions we used several (up to five) ESNs by training and running them in parallel, then averaging their outputs $y(t)$.

4. Results

We have obtained official scores in the Challenge with three of the above discussed algorithms: Table 1. The best scores are in bold with a mention of the place they have ranked in the Challenge at the time of writing. Event 4 is a mean squared error (MSE) scored in the domain of fetal heart rate (fHR) obtained from the annotations, and Event 5 is root MSE in raw fRR interval durations between the produced and the correct annotations of Dataset B [2].

Table 1. Challenge scores with different algorithms.

Algorithm	Event 4	Event 5
fRR stats (5)	**66.327** (7th)	11.027
fΔRR stats (7)	254.143	8.675
fRR +fΔRR stats (8)	147.236	**8.239** (5th)

An observation can be made that algorithms taking into account fRR statistics are less likely to wonder into improbable fHR values, while those using fΔRR statistics tend to track every fR more precisely (when at all) and thus perform better in the fRR domain.

5. Discussion

Our approach scored high among around 40 (?) other contestants. It is also quite fast: our slowest submission (8) took about 30 s to process one signal on a up-to-date personal computer, i.e., twice the real time. The algorithms could easily be made more precise sacrificing the speed, by making ESNs bigger and/or using more of them.

Using the statistics of the R events biases the system to a usual scenario: it improves correctness of annotations in healthy settings, but might be less suited to recognize abnormal events, which might be more important. This should be kept in mind, and emergency events prioritized if necessary by adapting the statistical models.

Our solution also did not produce the shape of fECG, but it could be done from the fR annotations and the pre-processed signal in the same way as for mECG. Alternatively, ESN could probably learn to extract fECG directly, given the reference training signals.

Our fQRS detection is also quite generic and could be used for other types of signals and annotations, without much hand-crafting because it uses machine learning and statistics estimated from data. In fact, our experiments showed, that fQRS detection using the method described in Section 3.3 is also possible directly on the signal coming from initial preprocessing (Section 3.1), without the removal of mECG described in Section 3.2, even though the latter helps to improve the performance, especially on difficult signals.

Acknowledgements

References

[1] Sameni R, Clifford GD. A Review of Fetal ECG Signal Processing; Issues and Promising Directions. The Open Pacing Electrophysiology Therapy Journal TOPETJ November 2010;3:4–20.

[2] Noninvasive Fetal ECG: the PhysioNet/Computing in Cardiology Challenge 2013. URL http://www.physionet.org/challenge/2013/.

[3] Martens SMM, Rabotti C, Mischi M, Sluijter RJ. A robust fetal ECG detection method for abdominal recordings. Physiological Measurement 2007;28(4):373.

[4] Jaeger H. The "echo state" approach to analysing and training recurrent neural networks. Technical Report GMD Report 148, German National Research Center for Information Technology, 2001.

[5] Jaeger H. Echo state network. Scholarpedia 2007;2(9):2330. URL http://www.scholarpedia.org/article/Echo_state_network.

[6] Lukoševičius M, Jaeger H. Reservoir computing approaches to recurrent neural network training. Computer Science Review August 2009;3(3):127–149. ISSN 1574-0137.

[7] Lukoševičius M. A practical guide to applying echo state networks. In Montavon G, Orr GB, Müller KR (eds.), Neural Networks: Tricks of the Trade, 2nd Edition, volume 7700 of LNCS. Springer. ISBN 978-3-642-35288-1, 2012; 659–686.

Address for correspondence:

Mantas Lukoševičius
Biomedical Engineering Institute, KTU
Studentu g. 65, LT-51369 Kaunas, Lithuania
mantas.lukosevicius@ktu.lt