Study of ECG Quality using Self Learning Techniques

Gianfranco Toninelli[1], Alfonso Gerevini[2], Ivan Serina[2], Martino Vaglio[1], Fabio Badilini[1]

[1]AMPS-LLC, New York, NY, USA
[2]DII, University of Brescia, Italy

Abstract

The aim of this study was to develop a method that could automatically evaluate the quality of ECG recordings.

In several situations, people performing the recording don't have the knowledge to evaluate the quality of the ECG and an immediate feedback about it would be useful.

Since there is not a consensus on how to define and quantify ECG quality, we applied self learning techniques starting from a set (N=1000) of randomly selected ECGs from our internal repository.

The full set of ECGs was blindly flagged by an expert cardiologist and subsequently analyzed by AMPS software which automatically computes a set of quality metrics. These quality parameters were used to train a neural network and build a decision tree.

The performance of the proposed solutions were evaluated using the mean squared error (MSE) between expected results (from the ECGs set) and obtained results (from neural network and decision tree).

The MSE resulting from the neural network and the decision tree were 0.01 and 0.004, respectively, indicating an error in range of 1%.

1. Introduction

The issue of the quality assessment of an ECG is not widely described in the literature, even though an immediate feedback would prove useful during the acquisition phase. In addition, it is not clear which quality measures are available and which ones are mostly used.

Some of the proposed methods are based on QRS detection[1], on noise quantification[2] and on signal mobility factors[3].

The aim of this paper is to introduce a method based on self-learning techniques, using a set of pre-evaluated ECGs to train a neural network (NN) and build a decision tree (DT).

2. Methods

A set of 1000 ECGs, with 12 leads and 10 seconds long, was randomly chosen from AMPS internal repository and each one of them was evaluated, in terms of quality, and blindly flagged as “good quality” or “bad quality” by an experienced cardiologist.

The set included 65% of the ECGs flagged as “good”, and the remaining 35% marked as “bad”.

The full set was subsequently analyzed by AMPS software (CalECG), which automatically computes a set of quality metrics.

The metrics considered for this study were: low frequency noise (LFNoise, obtained removing the high frequency band before assessing the noise level), high frequency noise (HFNoise, obtained removing the baseline wonder and the QRS complexes before computing the noise), an index representing the complexity of the repolarization signal (Tcomplex, obtained accounting number of zero-crossing of the first derivative of the repolarization section of the ECG signal) and a reliability index correlated with QRS detection performance of the embedded ECG measuring algorithm.

The last mentioned metric included 2 parameters: the difference between the number of detected QRS with the maximum and the minimum number of expected QRS peaks, DiffMax and DiffMin respectively.

The expected number of QRS was estimated from the Heart Rate (HR) measured during the recording process: number of maximum expected QRS = round(HR*10s/60+1); number of minimum expected QRS = round(HR*10s/60–1).

The noise metrics (LFNoise and HFNoise) have been computed performing a linear combination among the noise value of each of the 12 leads.

Each ECG was represented by five metrics and one status (0=good quality, 1=bad quality), an example is shown in Table 1.

Table 1. example of ECG metrics which represents the quality of an ECG, the noises values are represented in μV, the differences between detected and expected beats are represented in absolute value. Here 5 ECGs are reported (out of the overall 1000).

HFNoise	LFNoise	Tcomplex	DiffMax	DiffMin	status
3.51	10.8	1	0	2	0
3.12	8.04	1	1	1	0
5.40	8.57	0	1	1	0
3.34	8.26	2	1	1	0
7.56	14.9	1	1	3	1

The metrics were used as input of the self-learning techniques, and the status as the output (goal).

The two different adopted techniques (NN and DT) were evaluated using the mean squared error (MSE) between the results obtained with the self-learning methods and the goal results (status).

To build the two different self-learning methods, 80% of ECGs was involved in the training set (used to train the NN and build the DT), while the remaining 20% was only used in the test phase.

2.1. Training the neural network

The following step was to identify the structure of the NN to represent the ECG quality issue.

This problem is well represented by a continuous function, then we decided to implement a NN with one hidden layer (able to represent any kind of continuous function [4]).

For the reduced number of input and output nodes, 5 (one for each metric) and 1 (the goal status) respectively, we implemented a fully connected network.

The number of nodes used at the hidden layer was decided performing several tests, trying different configuration of the NN varying the number of hidden nodes, in this way we were able to find out the best configuration able to describe the ECGs quality issue.

The NN adopted was a feed forward network, using the Levenberg-Marquardt algorithm[5], which is a training algorithm based on back-propagation.

The large amount of examples in the training set allowed to further divide it to form an additional set, the validation set.

The validation set is generally used to preliminary validate the training process while it is in progress, giving a feedback on the generalization error and stopping the training process when this error increases (overfitting) [6].

We tested several configurations of the net changing the number of nodes at the hidden layer (10, 8 and 5), and the percentage of subdivision of the training and validation sets (40-40%, 50-30%, 55-25% and 60-20% of the total amount of examples).

For each test, the examples taken into account to populate the three different sets (training, validation and test) were randomly chosen from the starting set of 1000 ECGs.

For each configuration three different NN were trained, to examine the behaviour at each run of training. The configurations with oscillating results were discarded.

This method was implemented using MATLAB's neural network package and the results of the different configurations were compared.

2.2. Building the decision tree

As described above, to build the DT, the 80% of the ECGs from the starting set was involved and the remaining 20% was used during the test phase.

To build the decision tree we used Weka[7]; a tool designed ad hoc to build solutions to self-leaning problems.

The DT was built starting from the training set, following the standard C4.5[8], which involves the pruning of the non-significant leaves, each leaf representing a quality metric of the ECG.

The results obtained with this method were compared with the ones from the NN.

3. Results

The results regarding the several configurations of the NN are shown in Table 2.

Table 2. Performance of the different configurations of the NN: only the best performances (out of 3 runs) of each configuration are here reported.

	Hidden nodes	% training	% validation	MSE validation	MSE test
a)	10	40	40	0.005	0.013
b)	10	50	30	0.005	0.015
c)	10	55	25	0.001	0.011
d)	10	60	20	0.002	0.010
e)	8	40	40	0.001	0.029
f)	8	50	30	< 0.001	0.017
g)	8	55	25	0.004	0.015
h)	8	60	20	0.005	0.011
i)	5	40	40	0.0142	0.005
l)	5	50	30	0.0216	0.029
m)	5	55	25	0.016	0.005
n)	5	60	20	0.015	< 0.001

The configuration having 8 and 10 nodes in the hidden layer had no appreciable differences among the three runs performed. These configurations were considered stable.

The configuration with 5 hidden nodes presented differences among the runs. These differences are shown in Table 3.

Table 3. Performance of the three different NN training runs in presence of a hidden layer having 5 nodes.

Config.	Run	MSE validation	MSE test
i)	1	0.014	0.005
i)	2	0.022	0.029
i)	3	0.016	0.005
l)	1	0.015	< 0.001
l)	2	0.010	0.011
l)	3	0.022	0.029
m)	1	0.02	0.01
m)	2	0.016	0.032
m)	3	0.021	0.015
n)	1	0.022	0.013
n)	2	0.011	0.025
n)	3	0.019	0.041

The differences among the three runs for each configuration were visible and not negligible, indicating that the configuration with 5 nodes in the hidden layer is not stable and its behavior can change not only when the subdivision of the training set changes, but even when the same configuration is used. For this reason, the 5 nodes configuration was discarded.

The best performances were given by the configuration with 10 hidden nodes, having a MSE on the test set going from 0.01 to 0.015 in the different configurations, which indicates an error slightly greater than 1%. The MSE regarding the validation set went from 0.001 to 0.005, that indicates a very low presence of overfitting.

Furthermore, the implemented NN had 10 nodes at the hidden layer and was trained using a training set composed of 350 ECGs and a validation set composed of 250 ECGs. The performances were then tested using a test set of 200 ECGs.

Also the DT responded with good performances, with a MSE equals to 0.004 on the training set (composed by 800 ECGs).

Analyzing the DT, further information quality metrics can be assessed. During the pruning process of the decision tree, the Tcomplex metric was pruned, and then wasn't taken into account to evaluate the ECG quality. This could be due to the fact that this metric is not strictly related to the quality of an ECG, or that it does not give additional information compared to the other metrics.

In Figure 1, the structure of the DT is visible, indicating how the metrics are taken into account for the evaluation of the quality of an ECG.

The metrics related to the detection of the QRS were the most indicative: if the detection algorithm fails on detecting all the QRSs, the recording is most-likely presenting corrupted signal.

The other predictive metrics are HFNoise and LFNoise while, as described before, Tcomplex was not taken into account.

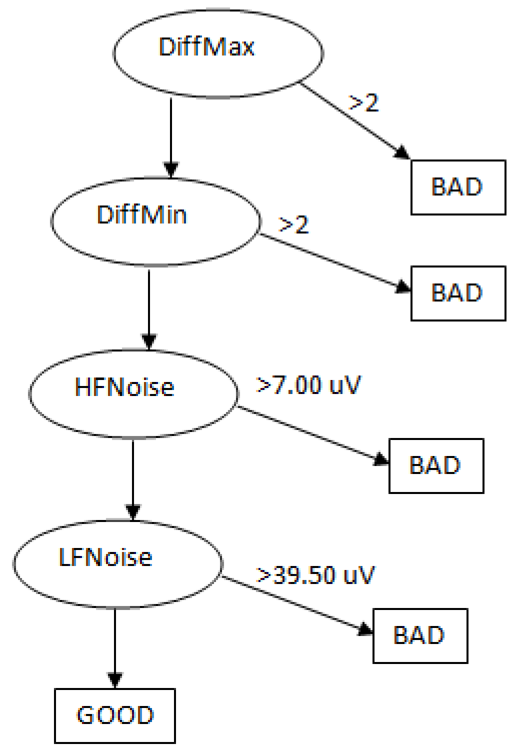

Figure 1. Structure of the decision tree built starting from a training set of 800 ECGs: the parameter Tcomplex was pruned, the other metrics were all taken into account to evaluate the ECG quality.

4. Conclusions

This preliminary study gave promising responses: the errors were acceptable (slightly higher than 1% using the NN and less than 1% using the DT) and some feedback about the correctness of the proposed metrics were obtained.

This model can be further strengthened, retraining the NN and rebuilding the DT using a larger set of ECGs and including other metrics which may better and heterogeneously represent the quality of an ECG.

The obtained results indicated that the proposed approach is viable and the example set should be enlarged, to obtain a stronger and more reliable model. Unfortunately this step would involve many hours of the cardiologist's work in the ECG flagging process.

With these preliminary results, this approach can be useful assessing the quality of ECGs in the context of data analysis of large quantity of ECGs, for example within multi-department clinical organizations and ultimately contribute to a better care.

References

[1] Hayn D, Jammerbund B. QRS detection based ECG quality assessment. Physiol Meas 2012;33:1449-1461.

[2] Johannesen L, Galeoti L. Automatic ECG quality scoring methodology: mimicking human annotators. Physiol Meas 2012;33:1479-1489.

[3] Haseri H, Homaeinezhad MR, Pourkhajeh H. An expert electrocardiogram quality evaluation algorithm based on signal mobility factors. J of Med Eng and Tec 2013;37:282-291.

[4] Hetch-Nielsen R. Kolmogorov's mapping neural network existence theorem. IEEE International Conference on Neural Networks 1987;3:11-13.

[5] Hagan MT, Menhaj MB. Training feed forward networks with the Marquardt algorithm. IEEE Trans on Neural Networks 1994;23:899-946.

[6] Sarle WS. Stopped training and other remedies for overfitting. Symposium on the Interface of Computing Science and Statistics 1995;27:352–360.

[7] Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH. The WEKA data mining software: an update. SIGKDD Explorations 2009;11:10-18.

[8] Quinlan JR. C4.5: Programs for Machine Learning. San Francisco: Morgan Kaufmann, 1993.

Address for correspondence.

Fabio Badilini, AMPS llc
Via P. Paolo VI, 34
25018 Montichiari (BS) – Italy

Phone: +39 030 9650745
Fax: +39 030 9650572
email : badilini@amps-llc.com