CinC Challenge - Assessing the Usability of ECG by Ensemble Decision Trees

Sebastian Zaunseder[1], Robert Huhle[1], Hagen Malberg[1]

[1]Dresden University of Technology, Institute of Biomedical Engineering, Dresden, Germany

Abstract

For various biomedical applications, an automated quality assessment is an essential but also complex task. Ensembles of decision trees (EDTs) have proven to be a suitable choice for such classification tasks. Within this contribution we invoke EDTs to assess the usability of ECGs. Our classification relies on the usage of simple spectral features which were derived directly from individual ECG channels. EDTs are generated by bootstrap aggregating while invoking the concept of random forrests. Though their simplicity, the trained ensemble classifiers turned out to be a very robust choice yielding an accuracy of 90.4 %. Therewith, the proposed method offers a good tradeoff bewteen accuracy and computational simplicity. Further improving the accuracy, however, turns out to be hardly feasible considering the chosen feature space.

1. Introduction

In ECG processing, the quality assessment of signals, or even signal excerpts, is an essential issue for telemedical applications as the one targeted by the CinC Challenge 2011. But even considering other areas it is often favorable to exclude signal portions from the analysis rather than deriving incorrect, eventually missleading, assumptions from those segments. A proper assessment of a signal's usability thus contributes to a successful automated ECG analysis.

Assessing the usability of ECGs constitutes a typical classification task. Decision trees (DTs) have been effectively used for regression and classification tasks. Simplicity when applying DTs, interpretability, the ability to handle missing attributes as well as the characteristic of being non-parametric are considered as main advantages of DTs. In contrast, DTs run a high risk of overfitting to training data and may require rather complicated structures to solve simple problems satisfactory [1]. Furtheron, considering classification tasks, it has been shown, that DTs usually do not reach the performance of more sophisticated approaches such as Support Vector Machines [2].

An attempt to improve the efficacy of modestly performing classifiers, so-called *weak learners* [3], consists in ensembling a number of them [4]. Using an ensemble of individual classifiers (called *base classifiers*) instead of a single individual classifier has not only shown to improve the results in comparison to the base classifier's performance, but even may outperform sophisticated classifiers [2]. This renders the application of ensemble methods an interesting approach for many classification tasks.

Unfortunately, improved results by invoking ensembles are gained at the expense of reduced interpretability and increased computational effort [5]. When considering DTs as base classifiers, though the ensemble implies an increased complexity, the computational effort is usually kept manageable as each of the base classifiers constitutes an exceptional easy classification paradigm. Depending on the application, the reduced interpretability, in turn, may constitute a significant drawback of applying ensembles. Just in the context of the CinC challenge interpretability, indeed, would be a nice feature in order to provide immediate assistance for the user to create an improved record when a first attempt failed.

Within this contribution we seek to exploit the strength of ensemble learners in order to create well-performing classifiers. By the way, we try to evaluate the possibility to obtain a simple classifier from a well working ensemble which complies with the desirable property of being interpretable.

2. Methods

2.1. Decision trees

When considering classification tasks, a decision tree T is a tree shaped classifier which consists of nodes t and edges. Any tree origins from a node without any incoming edge, called *root node*. The terminal nodes, i.e. nodes which do not possess any outcoming edges, are called *leaves*. The remaining nodes are called internal nodes. To each leaf a class or even a class probability is assigned. Each of the non-leave nodes represents a split regarding the input space. Such split is represented by a decision $\Phi(.)$. Most often, univariate decision, i.e. $\Phi(.) = \Phi(x)$ of the from "$x \geq threshold$" or "$x \in set$" where x represents a single attribute, are considered.

Growing a classification tree faces the task of recursive partitioning the input space. The input space is commonly represented by a learning set $\mathcal{L} = \{(\boldsymbol{x}_n, y_n) \,|n = 1, \ldots, N\}$ (also referred to as training data) consisting of N instances which are represented by a feature vector $\boldsymbol{x}$ and its belonging class y. To classify an instance $\boldsymbol{x}$ by a trained decision tree, i.e. making hypothesis $h(\boldsymbol{x})$ on the class membership of $\boldsymbol{x}$, the instance is propagated through the tree and assigned to the class to which the leaf belongs where the instance ends up.

Various implementations to train DTs have been described in the literature. Proposed variants vary as regards the criterion function which guides the tree growing, allowed splitting rules or the way in which trained trees are pruned. Amongst proposed variants, ID3 [6], C4.5 [7] and CART (classification and regression trees) [8] constitute prominent concepts to grow decision trees. Usually, such concepts can be understood as algorithmic frameworks which still exhibit some degrees of freedom when growing a DT.

2.2. Ensemble learners

Decision trees are considered as so-called *weak learners*, i.e. it is assumed that they may produce an hypothesis that performs only slightly better than random guessing [3]. Moreover, they have been shown to be unstable in that sense that small changes between learning sets $\mathcal{L}_1$ and $\mathcal{L}_2$ may result in considerable modifications in the resulting trees $T_{\mathcal{L}_1}$ and $T_{\mathcal{L}_2}$ [9]. Considering the former aspect, the need to improve the results of applying DTs is apparent. The latter, however, constitutes the basis for succesfully using ensembles instead of individual trees.

Ensemble methods build up their decision on the combined output of several individual classifiers: the most common methods of combination rely on simple majority voting or even weighted aggregation of the individual results [5]. Thereby, the diversity of its base classifiers is one of the cornerstones of efficient ensembles [4]. The most popular way to introduce diversity is to vary the training data. However, precondition to induce diversity is the previously refered instability of the base classifiers. Even to vary the learning data different methods have been outlined. Some of them vary the training data without considering the results of previous training iterations (e.g. Bagging (short for Bootstrap Aggregating) [9]), others do consider the devolution in the training (e.g. Boosting algorithms as AdaBoost [10]).

We decided to use Bagging as it was shown to be very robust even in the case of outliers. Through Bagging, variations are introduced into the ensemble by using bootstrap samples $\mathcal{L}_i^{BS}$ to train each of the base classifiers. To create a bootstrap sample $N_{BS} = \left|\mathcal{L}_i^{BS}\right|$ instances are drawn randomly from $\mathcal{L}$ with replacement [11]. This results in partially overlapping, but differing learning sets.

Beyond varying $\mathcal{L}$, even other methods are known to improve the performance of ensembles by increasing their diversity. Common approaches include the usage of variable classifier paradigms, the variation of classifier's inherited properties (e.g. randomly initialized weights in Multi Layer Perceptrons) or adapting the feature space. Considering DTs as base classifiers, these approaches have been denoted as Random-Forrests [12]. The latter approach, adapting the feature space, also is known as Random-Subspaces [13]. Random-Subspaces are implemented by considering just a subspace of the feature space $\mathcal{X}$ at each split when growing T. Random-Subspaces can be easily combined to bagging by which the accuracy of an ensemble is increased and the training effort is reduced at the same time [12].

3. Implementation

3.1. Features

Our choice of the features to be used, including the channels from which they are derived, relies on three constraints: (1) Each channel has same importance when a record is classified. (2) Inconsistencies regarding the orientation of a channel relative to the other channels are not considered as reason to discard a record. (3) The features should be elementary to reduce effort of calculation. From constraint 1 we derived that all channels somehow should be incorporated in the algorithm. Following constraint 2, the features can be extracted independently from all channels. Considering constraint 3, our classification relies exclusively on the usage of ordinary spectral features.

According to ECG's frequency contents (coarsely), we define the ranges $F_{SI} = 0.5\,\text{Hz} - 40\,\text{Hz}$, representing the most important signal information, $F_{LF} = 0.0\,\text{Hz} - 0.5\,\text{Hz}$, representing low frequency noise, and $F_{HF} = 45\,\text{Hz} - 250\,\text{Hz}$, representing high frequency noise. Each single signal channel is partitioned in 4 segments of equal length (2.5 s). From those segments and from the whole signal we extract the power in the previously defined frequency regions. Thereto, after windowing by using a Hamming window, the Fourier Transform is calculated and absolute values are extracted from the respective frequency bands. In addition to using mere powers, the power ratio of SI and LF power, and the power ratio of SI and HF power are extracted. These features are extracted from all channels. Therewith, all in all 300 features are derived per record (see table 1 for an overview).

Under constraint 1 the occurrence of a realization, not its belonging to one channel is of major importance. Its thus advantageous to sort the realizations of each feature by their ranks. To assess the quality of a record we do not consider all realizations but just the mean and the extrema regarding each feature to characterize that record. Thereby,

Table 1. Characterization of the used features where i indicates the used channel → $i = 1, 2, \ldots, 12$ and j the segment which is used if the signal is not considered as a whole → $j = 1, 2, 3, 4$.

Features	**Description**
P_i^{SI}, $P_{i,j}^{SI}$	Power in the signal band F_{SI}
P_i^{HF}, $P_{i,j}^{HF}$	Power in the high frequency band F_{HF}
$P_{i,j}^{LF}$, $P_{i,j}^{LF}$	Power in the low frequency band F_{LF}
$P_i^{SI\_HF}$, $P_{i,j}^{SI\_HF}$	Signal power to high-frequency power ratio → $P_i^{SI\_HF} = \frac{P_i^{SI}}{P_i^{HF}}$, $P_{i,j}^{SI\_HF} = \frac{P_{i,j}^{SI}}{P_{i,j}^{HF}}$
$P_i^{SI\_LF}$, $P_{i,j}^{SI\_LF}$	Signal power to low-frequency power ratio → $P_i^{SI\_LF} = \frac{P_i^{SI}}{P_i^{LF}}$, $P_{i,j}^{SI\_LF} = \frac{P_{i,j}^{SI}}{P_{i,j}^{LF}}$

the powers derived per segment are regarded as a single attribute and the following reduced feature set (F1 to F35) is derived:

$$\begin{array}{lll}
F1\text{: Max SI power} & := & \max_i \left(P_i^{SI}\right) \\
F2\text{: Mean SI power} & := & \frac{1}{12}\sum_i \left(P_i^{SI}\right) \\
F3\text{: Min SI power} & := & \min_i \left(P_i^{SI}\right) \\
F4\text{: Max segment SI power} & := & \max_{i,j} \left(P_{i,j}^{SI}\right) \\
F5\text{: Min segment SI power} & := & \min_{i,j} \left(P_{i,j}^{SI}\right) \\
F6\text{: Max HF power} & := & \max_i \left(P_i^{HF}\right) \\
 & \vdots & \\
F35\text{: Min segment SI-LF ratio} & := & \min_i \left(P_i^{SI\_LF}\right)
\end{array}$$

where i indicates the used channel, j the segment.

3.2. Classifier

The training of each single tree is covered by the framework which is established by CART. CART creates trees which just make use of binary splits. As splitting criterion we use the Gini-Index without considering other possibilities (justified as the used criterion function is assumed to have minor influence on the final results [1, 8]). As stopping criterion in the tree growing process we invoke a minimum number of instances N_{min} which has to be represented by each leave. To evaluate the influence of this number, N_{min} is variable with $N_{min} \in \{1, 6, 50\}$[1]. No pruning is done as the ability to generalize is a result of relying on the ensemble. To train tree T_i, a bootstrap sample $\mathcal{L}_i^{(BS)}$ of size $|\mathcal{L}| = 1000$ is drawn from $\mathcal{L}$. In each split we only consider $\sqrt{N} \approx 6$ attributes, thus implementing a Random Forrest. An ensemble E consists of 100 tress. This size was chosen as larger ensembles should not perform worse than smaller ones and the computational effort to train and run the ensemble is small, even with this big number of DTs [9]. The ensemble assigns the class by applying a mean rule

$$h_E(x) = \arg\max_j \sum_{i=1}^{N_{Trees}} h_{i,j}(x) \qquad (1)$$

where the hypothesis $h_{i,j}(x)$ is the support of the ith tree's winning leaf t^* to class j, i.e. $p(j|x)$. If $N_{min} = 1$ it holds $p(j|x) \in \{0, 1\}$. To allow some statistical evaluation and assess the influence of random drawing we trained 50 ensembles for each of the evaluated N_{min}.

3.3. Simplifying ensembles

As stated before, beyond mere classification we are interested in the possibility of simplifying a well-working ensemble in order to obtain an interpretable classifier. Thereto, an intuitive way is to create pruned versions of the base classifiers T_i, and evaluate their distance to the ensemble by the indicator function $I_{h_E,h_i}(x)$ where I equals 1 for $h_E(x) = h_i(x)$. The individual tree T_i to be chosen to represent the ensemble is the one which maximizes $\sum_{x \in SetA} I_{h_E,h_i}(x)$.

4. Results

Figure 1 shows the typical evolution of the out-of-Bag (ooB) error for ensembles with differing N_{min}. Table 2 gives an overview over the mean error on the training data (Set A), ooB errors and the results concerning Set B.

As the correct classifications for Set B are not available to date, the possibilities regarding the pruned classifiers are limited. However, by calculating the distance (1-0-loss) compared to our challenge entries an idea about the pruned classifier's performance can be obtained. The distance between two hypothesises h_1 and h_2 is calculated after $\sum_{x \in SetB} (1 - I_{h_1,h_2}(x))$. Figure 2 contains the distances.

[1] Actually more choices were evaluated, but the given ones are considered throughout the remaining article

Table 2. Classification errors (in %) for different N_{min}. Errors on Set A and ooB-Errors are given as $mean \pm sd$ (to Set B one ensemble for each N_{min} was applied).

N_{min}	**Set A**	**ooB**	**Set B**
1	0.21 ± 0.03	7.41 ± 0.33	10.4
6	3.4 ± 0.16	7.12 ± 0.29	9.6
50	5.9 ± 0.14	6.97 ± 0.24	9.8

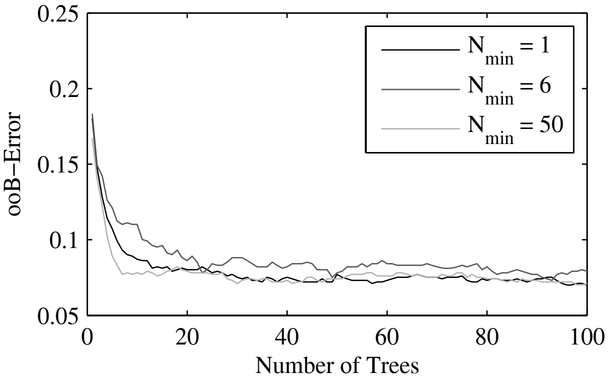

Figure 1. ooB-Errors for differing N_{min}.

5. Discussion and conclusions

As indicated by the final results of 90,4 %, the proposed method constitutes an reasonable solution. Although our first results were quite promising, to improve them, or even modify them noticeable, by invoking differing numbers of N_{min} turned out to be hardly possible. This holds, as stated by figure 2, though there are deviating class assignments in up to 25 records between our challenge entries. In average, those deviations cancel each other out. We understand these findings, as well as the outcome of some additional experiments which we could not report here, as a hint that an even more accurate classification is not feasible using our feature space. Considering the complexity of the 12 channel ECG, this outcome is reasonable as our feature space constitutes an extremly simplified approach.

Interestingly, also the single tree classifiers do not perform much different. Considering the indifferent influence of deviating classifications which was previously stated (see figure 2), even those differences may not necessarily mean a degradation of the results. This renders the usage of interpretable, individual classifiers rather interesting. However, as concerns the used features, throughout the individual classifiers there is no obvious tendency to use a certain feature.

To summarize, we rate our classification suitable considering its tradeoff between accuracy and computational simplicity (even more when considering the individual trees which perform rather similar to the ensembles). However, as accuracy is surely considered to be more important than simplicity, future research should attend to the possibility of an expanded feature space.

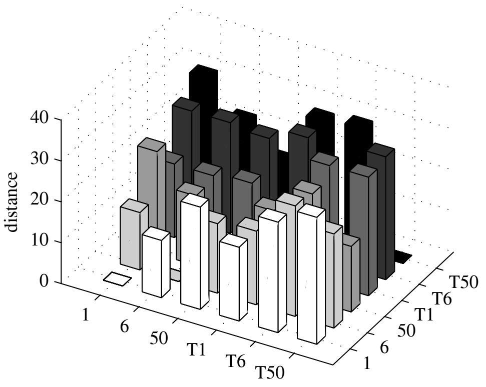

Figure 2. Pairwise distance between classifiers. x-axis and y-axis state the ensembles for diffferent N_{min} ($N_{min} = 1, 6, 50$) and individual trees $T1, T6, T50$ which were derived from these ensembles, respectively.

References

[1] Rokach L, Maimon O. Decision Trees. In The Data Mining and Knowledge Discovery Handbook. Springer, 2005; 165–192.

[2] Caruana R, Mizil AN. An empirical comparison of supervised learning algorithms. In Proceedings of the 23rd international conference on Machine learning, ICML '06. New York, NY, USA: ACM. 2006; 161–168.

[3] Schapire RE. The Strength of Weak Learnability. Machine Learning 1990;5:197–227.

[4] Polikar R. Ensemble based systems in decision making. IEEE Circuits and Systems Magazine 2006;6(3):21–45.

[5] Rokach L. Ensemble Methods for Classifiers. In The Data Mining and Knowledge Discovery Handbook. Springer, 2005; 957–980.

[6] Quinlan JR. Induction of decision trees. Machine Learning March 1986;1(1):81–106.

[7] Quinlan JR. C4.5: programs for machine learning. Springer Netherlands, 1993.

[8] Breiman L. Classification and regression trees. Chapman & Hall, 1993.

[9] Breiman L. Bagging Predictors. Machine Learning 1996;24:123–140.

[10] Freund Y, Schapire RE. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting,. Journal of Computer and System Sciences 1997;55(1):119 – 139.

[11] Efron B. Estimating the Error Rate of a Prediction Rule: Improvement on Cross-Validation. Journal of the American Statistical Association 1983;78(382):316–331.

[12] Breiman L. Random Forests. Machine Learning 2001;45:5–32.

[13] Ho TK. The random subspace method for constructing decision forests 1998;20(8):832–844.

Address for correspondence:

Sebastian Zaunseder
Dresden University of Technology, Institute of Biomedical Engineering, 01062 Dresden, Germany
sebastian.zaunseder@tu-dresden.de