Biomedical Signal Processing and Control 19 (2015) 130–136

Contents lists available at ScienceDirect

Biomedical Signal Processing and Control

journal homepage: www.elsevier.com/locate/bspc

Genetic algorithm-based method for mitigating label noise issue in ECG signal classification

Edoardo Pasolli[a,*], Farid Melgani[b]

[a] School of Civil Engineering, Purdue University, 47907 West Lafayette, IN, United States
[b] Department of Information Engineering and Computer Science, University of Trento, 38123 Trento, Italy

ARTICLE INFO

Article history:
Received 14 March 2014
Received in revised form
24 September 2014
Accepted 28 October 2014
Available online 27 November 2014

Keywords:
ECG signal classification
Genetic algorithms
Mislabeling issue
Multiobjective optimization
Training sample validation

ABSTRACT

Classification of electrocardiographic (ECG) signals can be deteriorated by the presence in the training set of mislabeled samples. To alleviate this issue we propose a new approach that aims at assisting the human user (cardiologist) in his/her work of labeling by removing in an automatic way the training samples with potential mislabeling problems. The proposed method is based on a genetic optimization process, in which each chromosome represents a candidate solution for validating/invalidating the training samples. Moreover, the optimization process consists of optimizing jointly two different criteria, which are the maximization of the statistical separability among classes and the minimization of the number of invalidated samples. Experimental results obtained on real ECG signals extracted from the MIT-BIH arrhythmia database confirm the effectiveness of the proposed solution.

1. Introduction

In the last decades, growing attention has been given in the biomedical engineering community to the problem of automatic analysis of electrocardiographic (ECG) signals. The great interest for ECG analysis derives from its role as an efficient and noninvasive tool for detecting and monitoring cardiac diseases. In particular, significant effort has been spent in the development of efficient and robust systems for ECG signal classification in order to detect automatically heartbeat abnormalities.

For such purpose, different solutions based on pattern recognition approaches have been proposed in the literature. Most of the attention has been given on improving the accuracy of the classification process by acting mainly at two different levels: (1) signal representation and (2) optimization of the discriminant function. In terms of signal representation different types of features have been extracted from the acquired ECG signals in order to have a better discrimination among the classes. Some examples of features are given by high-order statistics [1], morphological features [2], temporal intervals [2–4], wavelet transform coefficients [3–5], frequency domain features [6], and statistical features [7]. Moreover, given the high number of features that is usually involved, some feature reduction techniques have been applied to project the features into a lower dimensional feature space, such as principal component analysis [4,8] and independent component analysis [8]. The problem of discriminant function optimization has been addressed by considering different classification approaches. Although linear models have shown good results [2], in the last few years more attention has been given to nonlinear approaches. In particular, different works have focused on neural networks [3,4,8,9] and kernel methods such as support vector machines (SVMs) [1,5,7,8]. Moreover, classification improvements have been obtained by combining classifiers with optimization processes, such as those based on particle swarm optimization (PSO) [10,11].

Although these works have demonstrated their effectiveness, they are based on an essential assumption that is the samples used to train the classifier are statistically representatives of the classification problem to solve. Therefore the quality and the quantity of such samples are very important, because they have a strong impact on the performance of the classifier. However, the process of training sample collection is not trivial since it is based on a human user (cardiologist) intervention and so it is subject to errors and costs both in terms of time and money. In general, scarce attention has been given to this problem in the literature. Only in the last few years there has been a growing interest in developing semi-automatic strategies for the problem of training set construction.

* Corresponding author. Tel.: +1 7654097937.
E-mail addresses: epasolli@purdue.edu (E. Pasolli), melgani@disi.unitn.it (F. Melgani).

http://dx.doi.org/10.1016/j.bspc.2014.10.013
1746-8094/

A first problem is given by the scarcity of available training samples due to complexity and cost that characterize the training sample collection process. Accordingly, this constrains the classification process to be carried out with a small number of training samples, thus leading to weak estimates of the classifier parameters and potentially bad classification performances, in particular if class distributions are overlapped. A solution to this problem is represented by active learning [12], which has been proposed recently for ECG signal classification [13,14]. Considering a small and suboptimal initial training set, few additional samples are selected from a large amount of unlabeled data. These samples are labeled by the human user and then added to the training set. The entire process is iterated until a stopping criterion is satisfied. The aim of active learning is to rank the learning set according to an opportune criterion that allows to select the most useful samples to improve the model, thus minimizing the number of training samples necessary to maintain discrimination capabilities as high as possible.

Another problem in real application scenarios is represented by the mislabeling issue due to errors in the process of sample labeling. Since the presence of mislabeled training samples has a direct negative impact on the classification process, the development of automatic techniques for validating the collected samples is crucial. Few solutions for coping with this issue have been proposed in the machine learning community. They are based on two main approaches. The first one admits the presence of mislabeled samples, but aims at designing a classifier that is less influenced by this presence [15]. The second one tends to identify and remove the mislabeled samples from the training set [16–20]. The process of mislabeled sample removing was done by considering different classification approaches, such as k-nearest neighbors (kNNs) [16], C4.5 [17], and classifier ensemble [18], or by adopting geometrical graph theory [19]. In [20], a clustering technique based on expectation maximization algorithm was used to estimate for each training sample a probability vector of class membership. The confidence on the current label was used as a weight during the construction of the classification model. Although the promising performance exhibited by these approaches, to the best of our knowledge the problem of mislabeled samples has been not considered in the context of ECG signal classification.

The objective of this paper is to investigate the problem of training sample validation for ECG signal classification. In particular, the proposed approach takes inspiration from the strategy proposed in [21], in which the mislabeled sample detection issue was viewed as an optimization problem where it was looked for the best subset of training samples. This strategy, proposed specifically for classification of remote sensing images, supposes that classes follow a Gaussian distribution. Although this assumption is often verified in the remote sensing context, it is not true in the case of ECG signals. For this reason a more general method, i.e., applicable also to non-Gaussian distributions, is proposed in this work. The optimization problem is formulated within a genetic algorithm (GA) [22,23]-based framework, thanks to its capability to solve complex pattern recognition problems [24,25]. Each chromosome is configured as a binary string, which represents a candidate solution for validating/invalidating the available training samples. The genetic optimization process consists of optimizing jointly two different criteria, which are the maximization of the statistical separability among classes and the minimization of the number of invalidated samples. The former is based on *k*NN classification. The latter allows to get at convergence a Pareto front from which the human user can select the best solution according to his/her prior confidence on the reliability of the collected training set.

The proposed approach is validated experimentally on real ECG signals from the well-known MIT-BIH arrhythmia database [26]. The obtained results show that the proposed automatic validation strategy is able to detect the mislabeled samples with a high accuracy. Moreover, the removal of the detected mislabeled samples has a good impact on the accuracy given by the state-of-the-art support vector machine (SVM) [27] classification.

The rest of the paper is organized as follows. In Section 2, we summarize the basic idea of the multiobjective genetic algorithm and describe the proposed strategy for automatic training sample validation. Section 3 presents the experimental results obtained on real ECG signals. Finally, conclusions are drawn in Section 4.

2. Proposed method

2.1. Problem formulation

Let us consider a training set L of ECG signals composed of n samples labeled by the human user (cardiologist). Each sample is represented by a vector of d features $\mathbf{x}_i \in \Re^d$ ($i = 1, ..., n$) and the corresponding label y_i. y_i assumes one of T discrete values, where T is the number of classes. The objective is to detect in an automatic way which of these n training samples are mislabeled in order to remove them from the training set before constructing the final classification model.

A simple approach to this problem would consist of trying all possible combinations of validated/invalidated training samples and then choosing the best one according to some predefined criteria. However, this appears computationally prohibitive and thus an impractical solution, since the number of possible combinations is equal to 2^n. An alternative consists of adopting an optimization process in order to find the best solution in the solution space. In this work, we propose to reach this objective by means of a multiobjective genetic optimization method. In the following subsections, we first introduce GAs. Then, after describing its two main components (i.e., the chromosome structure and the fitness function), we explain the different phases of the proposed genetic solution. The flow-chart of the proposed method is shown in Fig. 1.

2.2. Genetic algorithms

GAs are optimization techniques inspired from biological principles [22,23]. A genetic optimization algorithm performs a search by evolving a population of candidate solutions (individuals) modeled with chromosomes. The population is improved during the iterative process using genetic mechanisms based both on deterministic and nondeterministic operators. A traditional GA algorithm involves the following steps: (1) an initial population of chromosomes is generated randomly; (2) the goodness of each chromosome is evaluated according to a predefined fitness function representing the considered objective function; (3) the best and the worst chromosomes are kept and discarded, respectively, using an appropriate selection rule based on the principle that the better the fitness, the higher the chance of being selected; (4) the population is reproduced using genetic operators such as crossover and mutation; (5) the entire process is iterated until a defined convergence criterion is reached.

Several optimization problems require optimizing more than one fitness function simultaneously. This operation is not trivial since multiple measures of competing objectives have to be considered at the same time. This is referred as to multiobjective optimization problem. From a methodological point of view, multiobjective optimization consists of finding the solution that optimizes the ensemble of Q objective functions

$$f(\mathbf{p}) = [f_i(\mathbf{p}), i = 1, 2, ..., Q] \quad (1)$$

where $\mathbf{p}$ is a solution to the considered optimization problem. Different multiobjective GA-based approaches have been proposed in the literature [23]. In this paper, we consider the nondominated sorting genetic algorithm (NSGA-II) [28] for its low computational

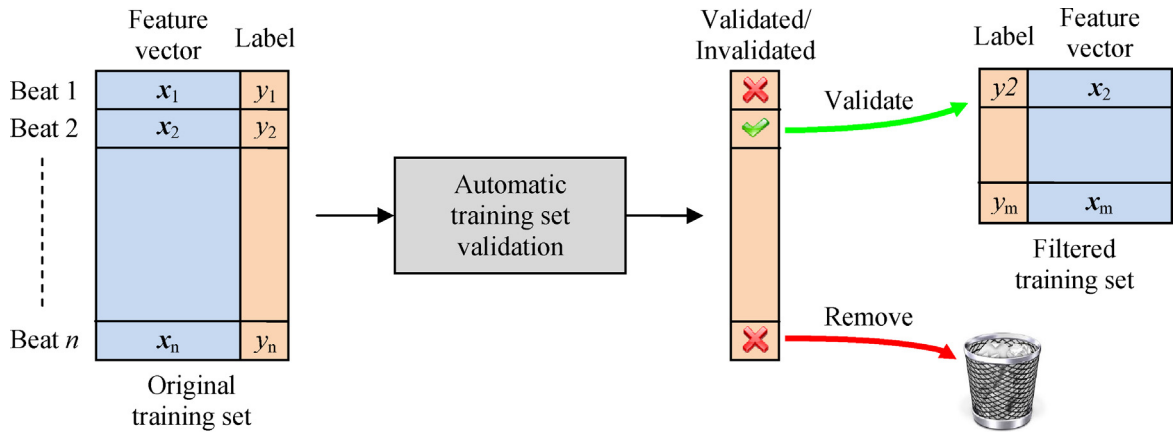

Fig. 1. Flow chart of the proposed automatic training sample validation framework.

requirements and its ability to distribute uniformly the solutions along the Pareto front. It is based on the concept of dominance which states that a solution $\mathbf{p}_i$ is said to dominate another solution $\mathbf{p}_j$ if and only if

$$\forall\, k \in \{1, 2, \ldots, Q\},\ f_k(\mathbf{p}_i) \le f_k(\mathbf{p}_j) \wedge \exists\, k \in \{1, 2, \ldots, Q\} : f_k(\mathbf{p}_i) < f_k(\mathbf{p}_j) \tag{2}$$

This concept leads to the definition of *Pareto optimality*: a solution $\mathbf{p}_i^* \in \Omega$ (Ω is the solution space) is said to be Pareto optimal if and only if there exists no other solution $\mathbf{p}_j \in \Omega$ that dominates $\mathbf{p}_i^*$. The latter is said to be *nondominated*, and the set of all nondominated solutions forms the *Pareto front* of optimal solutions. The algorithm can be summarized by the following steps: (1) an initial parent population of chromosomes is generated randomly; (2) the chromosomes selected through a crowded tournament selection undergo crossover and mutation operations to form an offspring population; (3) both offspring and parent populations are combined and sorted into fronts of decreasing dominance (rank); (4) the new population is filled with solutions of different fronts starting from the best one; (5) if a front can only partially fill the next generation, crowded tournament selection is used again to ensure diversity; (6) the algorithm creates a new offspring population and the process continues up to convergence.

2.3. Genetic algorithm setup

The use of GAs requires setting two main ingredients, i.e., the chromosome structure and the fitness functions, which translate the considered optimization problem and guide the search toward the best solution, respectively.

In our context, since the objective is to validate/invalidate each of the n available training samples, we consider a population of N chromosomes $\mathbf{C}_m$ ($m = 1, 2, \ldots, N$), where each chromosome $\mathbf{C}_m \in \{0, 1\}^n$ is a binary vector of length n that encodes a candidate combination of validated/invalidated samples. As shown in Fig. 2, a gene takes value "0" or "1" if the corresponding sample is validated or invalidated, respectively.

The validation/invalidation procedure is based on the hypothesis that mislabeling a training sample potentially leads to an increase of the intra-class variability. In [21], this intra-class variability increase was quantified as decrease of the between-class distance. For this purpose, the Jeffries–Matusita (JM) statistical

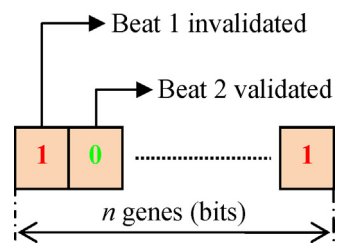

Fig. 2. Illustration of the chromosome structure.

distance measure [29] was adopted. The JM distance between two generic classes ω_i and ω_j is defined as

$$JM_{ij} = \sqrt{2(1 - e^{-B_{ij}})} \tag{3}$$

where B_{ij} is the Bhattacharyya distance measure. The JM measure is bounded by the interval $[0, \sqrt{2}]$. In particular it is equal to zero when the classes are completely overlapped and takes the value $\sqrt{2}$ when they are totally separated. The calculation of the Bhattacharyya distance for generic class distributions is computationally intensive, but it becomes tractable if Gaussian distributions are considered. In this case the B_{ij} measure is given by

$$B_{ij} = \frac{1}{8}(\mathbf{u}_i - \mathbf{u}_j)^T \left[\frac{\Sigma_\mathbf{i} + \Sigma_\mathbf{j}}{2}\right]^{-1} (\mathbf{u}_i - \mathbf{u}_j) + \frac{1}{2} \ln \frac{\left|\frac{\Sigma_\mathbf{i} + \Sigma_\mathbf{j}}{2}\right|}{\sqrt{\left|\Sigma_\mathbf{i}\right| \left|\Sigma_\mathbf{j}\right|}} \tag{4}$$

where $\boldsymbol{\mu}$ and $\sum$ denote mean vector and class covariance matrix, respectively. In this paper the intra-class variability is evaluated in a different way in order to deal with non-Gaussian distributions. The proposed strategy is based on kNN classification [29]. For a generic (validated) training sample $\{\mathbf{x}_l, y_l\}$ the accuracy score a_l is defined as

$$a_l = \frac{1}{k} \sum_{j=1}^{k} \delta_l^j \tag{5}$$

where $\delta_l^j = 1$ if the label y_l corresponds to the label of the jth nearest (validated) neighbor, otherwise it is equal to zero. The final intra-class variability measure A is obtained by averaging the scores a_l of the (validated) training samples. The A measure is bounded by the interval [0, 1], where the value one is verified when classes are totally separated. In order to reduce the computational load, the validation process is not performed on the original features, but on a reduced feature space obtained by applying principal component analysis [29]. In particular, the first five components are considered in our case.

At this point the optimization process needs an information from the human user, which is the expected amount of mislabeled training samples. Without this information, the process would validate only few samples per class, i.e., the most distant from the other classes. With this information, a first solution consists of running a constrained optimization process, which at convergence would provide the best subset of validated samples with the pre-specified number of mislabeled samples. However, this approach requires setting a priori the exact number of mislabeled samples. An alternative and smarter solution consists of considering a multiobjective optimization process based on the NSGA-II algorithm. In this case, we define a second fitness function that represents simply the number of invalidated samples. At convergence we obtain a Pareto front of different solutions from which the human user can select one solution according to his/her prior confidence on the reliability of the collected training sample set.

2.4. Algorithmic description

In the following we summarize the different phases that characterize the automatic training sample set validation method.

2.4.1. Phase 1: decomposition from multiclass to binary classification problems

The typical multiclass nature of the training set makes it necessary to resort to a suitable multiclass validation strategy. This is done by (1) decomposing the multiclass problem into $T(T-1)/2$ binary classification tasks (one-against-one strategy); (2) performing all "binary" genetic runs, i.e., running a genetic optimization process for each binary training set; (3) computing at convergence a validation/invalidation score function for each sample from the solutions provided by all binary runs; and (4) invalidating a sample if its "invalidation" score is greater than the "validation" one (winner-takes-all decision rule).

2.4.2. Phase 2: optimization with NSGA-II

This phase is described for a single binary genetic run, although it is similar for all runs.

- Phase 2.1: initialization

  Step 1) Generate randomly a population $P^{(t)}$ ($t = 0$) of N chromosomes $\boldsymbol{C}_m$ ($m = 1, 2, \ldots, N$), each gene taking either a "0" or a "1" value.

  Step 2) For each candidate chromosome $\boldsymbol{C}_m$ ($m = 1, 2, \ldots, N$) of $P^{(t)}$, build a new training set by removing from the original binary training set the samples invalidated by the corresponding genes (i.e., those with a "1" value) and compute its fitness functions (i.e., its intra-class variability measure A and number of invalidated samples).

  Step 3) Perform random binary tournament selection, crossover, and mutation operations in order to create a population of offspring $Q^{(t)}$ having the same size N of the population of the parents $P^{(t)}$.

- Phase 2.2: optimization

  Step 4) Merge the two populations, i.e., $R^{(t)} = P^{(t)} \cup Q^{(t)}$, for guaranteeing elitism (mechanism which ensures that all the best chromosomes are passed to the next generation), and thus stability and fast convergence of the optimization process. Sort the merged population R_t into different fronts of descending domination rank according to the nondominated sorting method.

  Step 5) Create a new generation $P^{(t+1)}$ of size N by choosing the first best N solutions from $R^{(t)}$. The last solutions of the same front are selected so that they span as much as possible their front. This is carried out by integrating in the selection procedure a crowding distance. This last is computed basing on the two solutions surrounding the solution under consideration in the performance space (i.e., the space defined by the two fitness functions). It plays a key role in a multiobjective optimization process since it permits to force it in obtaining final solutions which are as spread as possible along the Pareto optimal front.

  Step 6) If the stop criterion (e.g., maximal number of generations and/or a check on the variation of its intra-class variability measure A during the current and last generations) is not satisfied, set $t \leftarrow t + 1$ and go to *Step* 2.

2.4.3. Phase 3: sample validation

Step 7) Basing on the indication from the human user about his/her prior confidence on the reliability of the original training sample set, select from the Pareto front of each binary genetic run (i.e., couple of classes ω_i and ω_j) the chromosome $\mathbf{C}^*_{mij}$ with a number of invalidated training samples closest to this indication.

Step 8) For each training sample $\mathbf{x}_l$ ($l = 1, 2, \ldots, n$), compute a score function:

$$S(\mathbf{x}_l) = \sum_{i=y_l; j \neq i} \mathbf{C}^*_{mij}(l) \tag{6}$$

where y_l is the original label assigned by the human user and $\mathbf{C}(l)$ denotes the lth gene of the considered chromosome. Validate $\mathbf{x}_l$ if

$$S(\mathbf{x}_l) \leq (T-1)/2 \tag{7}$$

otherwise, remove it.

The human user confidence plays an important role in the last phase of the method. In general, if the confidence value is underestimated (precautionary behavior), this will result in a larger number of detected mislabeled samples. Conversely, if it is overestimated, the risk of not detecting part of actually mislabeled samples increases.

3. Experiments on real ECG signals

3.1. Dataset description and experimental setup

The method proposed for automatic training sample set validation was tested experimentally on real ECG signals, obtained from the well-known MIT-BIH arrhythmia database [26]. In particular, the considered beats referred to the following six classes: normal sinus rhythm (N), atrial premature beat (A), ventricular premature beat (V), right bundle branch block (RB), paced beat (/), and left bundle branch block (LB). The beats were selected from the recordings of 20 patients, which corresponded to the following files: 100, 102, 104, 105, 106, 107, 118, 119, 200, 201, 202, 203, 205, 208, 209, 212, 213, 214, 215, and 217. As done in [10], [13], the most predominant classes and less noisy recordings were considered in the experimental analysis. In order to feed the classification process, we adopted a subset of the features described in [2]: (1) ECG morphology features and (2) three ECG temporal features, i.e., the QRS complex duration, the RR interval (the time span between two consecutive R points representing the distance between the QRS peaks of the present and previous beats), and the RR interval averaged over the ten last beats. These features were extracted by (1) applying the *ecgpuwave* software [30] in order to detect QRS and recognize ECG wave boundary; (2) extracting the three temporal features of interest; and 3) normalizing the duration of the segmented ECG cycles to the same periodic length according to the procedure reported in [31]. For this purpose, the mean beat period was chosen as the normalized periodic length, which was represented by 300 uniformly distributed samples. Consequently, the total number of morphology and temporal features was equal to 303. Fig. 3 illustrates the distribution of the six considered classes in the subspace given by the

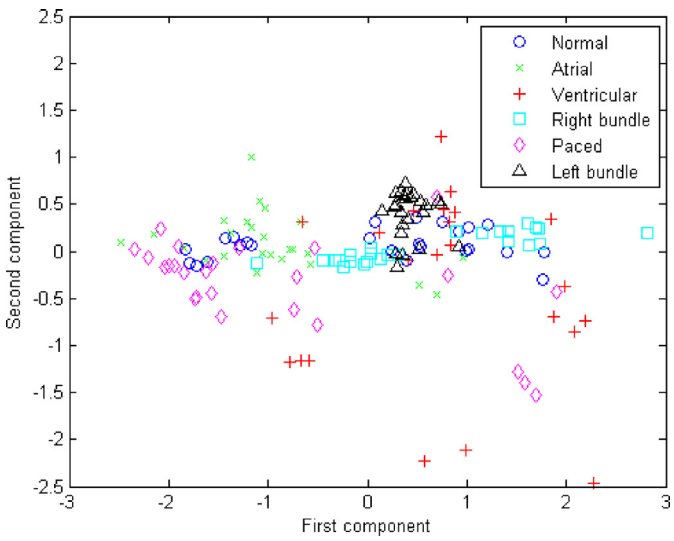

Fig. 3. Distribution of the six considered classes in the subspace given by the two first principal components. For better visualization, just 25 samples were randomly selected for each class.

Table 1
Number of training and test ECG beats used in the experiments.

Class	N	A	V	RB	*l*	LB	Total
# Training beats	75	50	50	25	25	25	250
# Test beats	24,000	238	3939	3739	6771	1751	40,438

two first principal components. A strong overlap among class distributions can be observed. Moreover, Kolmogorov–Smirnov test determined the non-Gaussianity of the distributions, as assumed in the introductive and methodological parts of the paper. All available samples were randomly split into two sets, corresponding to training and test sets. The training samples were used to apply the mislabeled sample detection method, while the test set was considered to evaluate generalization capabilities of the obtained classification model. The detailed number of samples is reported in Table 1.

We considered various training sample set validation scenarios by adding noise (i.e., mislabeling) with different proportions (i.e., 5%, 10%, and 20%) to the original noise-free (i.e., without any mislabeled sample) dataset. Mislabeling was carried out by permuting the label of randomly selected training samples. This allowed us to create a controlled experimental environment useful to understand how noise affects our approach. In all experiments, we set empirically the parameters for the genetic optimization process to these standard values: population size $N = 100$; maximum number of generations set to 500; crossover probability $p_c = 0.9$; and mutation probability $p_m = 0.01$. We note the population size and the maximum number of generations have a direct impact on the computational load and thus need to be kept relatively small in order to avoid an excessive execution time. The parameter k for computing the fitness function based on kNN classification was set to 5. At convergence of the genetic optimization process, we selected from the Pareto front the solution closest to the applied mislabeling rate. The assessment was done in terms of detection of mislabeled samples. In particular, we considered probability of detection P_D and false alarm P_{FA}. This latter gives information about the number of invalidated noiseless samples, while the former expresses the number of correctly invalidated mislabeled samples. All the experiments were repeated five times, each by selecting and mislabeling the samples in a random way, so that to yield statistically reliable results.

Table 2
Detection performance in terms of probability of detection (P_D) and false alarm (P_{FA}) achieved on the ECG beats versus the proportion of mislabeled training samples.

	GA-JM method Mislabeling proportion			GA-kNN method Mislabeling proportion		
	5%	10%	20%	5%	10%	20%
# Mislabeled samples	13	25	50	13	25	50
# Invalidated samples	28.80	67.20	86.60	12.60	28.20	45.80
P_D	63.07	74.40	77.20	78.46	78.40	72.40
P_{FA}	71.34	71.65	54.10	31.05	15.65	4.58

In addition, we evaluated the impact of the removal of the detected mislabeled samples on the accuracy given by a SVM classifier [27] based on the RBF Gaussian kernel. For each classification scenario, parameter tuning was carried out empirically by means of a K-fold cross-validation procedure ($K = 5$) performed on the training samples associated with the scenario. In cross-validation, the original training set is randomly subdivided into K equal-size subsets. $K - 1$ subsets are used as real training data and the remaining subset is retained for validating the model. The process is repeated K times and the K obtained results are averaged to produce a single estimation. Then, classification performances were evaluated on the test set in terms of different measures: (1) the overall accuracy (Acc), which is the percentage of correctly classified samples among all the considered samples, independently of the classes they belong to; (2) the specificity (Sp) related to class N, which is defined as Sp = TN/(TN + FP), and the sensitivity (Se) related to classes A, V, RB, *l*, LB, which is defined as Se = TP/(TP + FN), where TP: true positive, TN: true negative, FP: false positive, FN: false negative; (3) the average accuracy (AvAcc), which is the average over the Sp and the five values of Se.

3.2. Experimental results

In the first part of the experiments we evaluated the performance of the proposed automatic training set validation method in terms of detection of mislabeled samples. In particular, we compared the new proposed fitness function based on kNN classification (GA-kNN) with the strategy that adopts the JM statistical distance measure (GA-JM). The results in terms of number of invalidated samples, probability of detection P_D and false alarm P_{FA} are reported in Table 2. In general, better performance was exhibited by the GA-kNN strategy. In terms of number of invalidated samples the proposed method gave solutions closest to the real number of mislabeled samples. For example, considering a mislabeling proportion equal to 5%, i.e., 13 mislabeled samples, the GA-kNN method invalidated on the average 12.60 samples, while a greater number of samples (28.80) were detected by the GA-JM method. This aspect influences directly the performances in terms of P_D and P_{FA}. In particular, the results in terms of P_D were comparable, but much better values were obtained by the proposed solution in terms of P_{FA}. Considering a mislabeling proportion equal to 5%, P_D (P_{FA}) were equal to 78.46 (31.05) and 63.07 (71.34) for the GA-kNN and GA-JM methods, respectively. We note the large number of samples invalidated for the GA-JM strategy guaranteed to have good capabilities in terms of detection of mislabeled samples, but at the same time a high number of noiseless samples were wrongly invalidated.

In the second part of the experiments we evaluated the impact of the removal of the invalidated samples on the accuracy given by SVM classification. The results obtained for all the different considered training set scenarios are summarized in Fig. 4 and Table 3. Three cases can be taken as reference: (1) "Noise-free", (2) "Noisy" (represented with black line in Fig. 4 and Fig. 3) "Filtered-ideal" (blue line). "Noise-free" is the case in which all the 250 training beats were used without any mislabeling (i.e., with a mislabeling

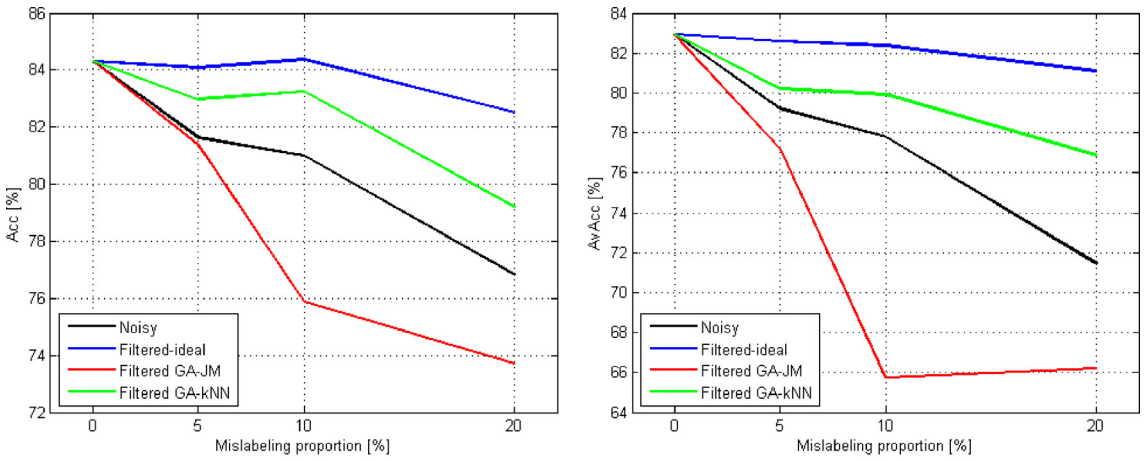

Fig. 4. (a) Acc and (b) AvAcc obtained by the SVM classifier on the ECG beats for all the considered training set scenarios. (For interpretation of the references to color in the text of this figure citation, the reader is referred to the web version of this article.)

Table 3
Acc AvAcc, Sp, and Se obtained by the SVM classifier on the ECG beats for all the considered training set scenarios.

Training set	Mislabeling proportion	Acc	AvAcc	Sp	A	V	RB	*l*	LB
Noise-free	0%	84.31	82.92	88.95	82.35	88.95	88.12	62.48	86.69
Noisy	5%	81.63	79.21	87.72	80.84	85.97	81.40	57.67	81.64
	10%	80.99	77.78	87.16	74.95	84.63	80.70	57.11	82.13
	20%	76.84	71.44	86.60	76.97	84.51	71.96	43.52	65.11
Filtered-ideal	5%	84.08	82.58	88.62	81.61	87.58	88.65	63.00	85.97
	10%	84.37	82.38	88.69	79.24	86.78	88.11	65.27	86.20
	20%	82.51	81.08	86.77	79.41	84.84	85.92	63.36	86.20
Filtered GA-JM method	5%	81.37	77.16	88.60	81.17	88.38	76.66	56.84	71.31
	10%	75.89	65.75	90.02	73.52	88.73	49.40	38.41	54.43
	20%	73.72	66.20	86.69	73.61	78.84	54.61	37.25	66.18
Filtered GA-*k*NN method	5%	82.99	80.22	88.25	77.39	86.03	85.91	61.81	81.91
	10%	83.25	79.93	88.79	76.30	80.58	86.06	63.62	84.22
	20%	79.21	76.89	85.32	76.33	82.82	78.23	54.91	83.73

proportion equal to 0%) to construct the classification model. In this situation we got Acc (AvAcc) equal to 84.31 (82.92), which represent an accuracy upper bound. Part of the training samples was mislabeled before classification in the "Noisy" case. As expected, a decrease of accuracies was obtained by increasing the number of mislabeled samples. For example, Acc (AvAcc) was equal to 81.63 (79.21) and 76.84 (71.44) for a mislabeling proportion equal to 5% and 20%, respectively. This is a lower bound of the accuracies since the classification model was constructed from the corrupted training set without applying any sample validation procedure. The third reference scenario is represented by "Filtered-ideal", in which all the mislabeled samples were removed from the training set before classification. This is an ideal situation that would correspond to the case in which the validation strategy is able actually to detect and remove all the samples wrongly labeled. In this case P_D and P_{FA} would be equal to 100% and 0%, respectively. We note removing of mislabeled samples guaranteed a good improvement in terms of classification accuracies with respect to the "Noisy" case, in particular when the mislabeling proportion was high. For example, Acc (AvAcc) was equal to 84.08 (82.58) and 82.51 (81.08) for a mislabeling proportion equal to 5% and 20%, respectively. In absolute terms, a slight accuracy decrease was verified at the increasing of mislabeling proportion since a greater number of samples were removed from the training set before classification. Finally, we report the results obtained by the proposed automatic validation strategy. In particular, the proposed GA-*k*NN and the reference GA-JM methods are reported in Fig. 4 with green and red lines, respectively. It is evident how the strategy based on JM distance, which supposes that classes are Gaussian distributed, is not appropriate for classification of ECG signals. It exhibited poor performance, even worse than the "Noisy" case. This is due to the fact that several correctly labeled samples were erroneously invalidated thus decreasing the number of good samples used to construct the classification model. Much better results were obtained by the proposed method based on *k*NN classification, which does not assume any a priori class distribution. Although the strategy did not reach the accuracies given by the "Filtered-ideal" case, a good improvement with respect to the "Noisy" scenario was verified. For example, Acc (AvAcc) was improved by 1.36 (1.01) and 2.37 (5.45) for a mislabeling proportion equal to 5% and 20%, respectively. We can conclude that the proposed method is effective to limit the negative impact of the mislabeled samples on the classification process even in situations where their presence in the training set is significant.

4. Conclusion

In this paper, we have introduced in the biomedical engineering community the problem of training sample validation for classification of ECG signals. The objective is to assist the human user (cardiologist) in his/her work of labeling by removing in an automatic way the training samples with potential mislabeling problems. For this purpose a new strategy based on genetic algorithms has been proposed. The experimental results obtained on real ECG signals confirm the effectiveness of the proposed solution:

(1) training sample mislabeling affects classifier design and performance since it has a direct impact on the class distributions. This problem is strictly related to the amount of mislabeled samples; (2) the proposed validation method is effective to limit the propagation of errors related to mislabeled samples in the signal classification framework even in cases where their presence is significant; (3) the proposed method does not assume any a priori class distributions thus resulting particularly suitable for ECG signal classification. Significant improvements have been verified with respect to similar strategies that suppose Gaussian distributed classes. Moreover, we note that the proposed method acts as a filter completely independent from the classification approach adopted in the classifier design phase.

The main drawback of the proposed solution is related to the computational load. The proposed algorithm required about 15 min to filter the training set composed by 250 labeled samples and hence can be applied in an off-line scenario. However, this problem can be alleviated by recurring to a parallel implementation of genetic algorithms. Moreover, the computational time can be heavy in presence of large size training set. We can deal with this issue by splitting the original training set into several training subsets and then processing each of them separately. We note that the proposed method has been tested on samples mislabeled in an artificial way. An experimental analysis on real mislabeld samples would be useful in order to validate further the investigated approach.

References

[1] R.S. Osowski, T.H. Linh, T. Markiewicz, Support vector machine-based expert system for reliable heart beat recognition, IEEE Trans. Biomed. Eng. 51 (April (4)) (2004) 582–589.

[2] P. de Chazal, M. O'Dwyer, R.B. Reilly, Automatic classification of heartbeats using ECG morphology and heartbeat interval features, IEEE Trans. Biomed. Eng. 51 (July (7)) (2004) 1196–1206.

[3] T. Inan, L. Giovangrandi, J.T.A. Kovacs, Robust neural network based classification of premature ventricular contractions using wavelet transform and timing interval features, IEEE Trans. Biomed. Eng. 53 (December (12)) (2006) 2507–2515.

[4] T. Ince, S. Kiranyaz, M. Gabbouj, A generic and robust system for automated patient-specific classification of ECG signals, IEEE Trans. Biomed. Eng. 56 (May (5)) (2009) 1415–1426.

[5] A. Daamouche, L. Hamami, N. Alajlan, F. Melgani, A wavelet optimization approach for ECG signal classification, Biomed. Signal Process. 7 (July (4)) (2012) 342–349.

[6] C.-H. Lin, Frequency-domain features for ECG beat discrimination using grey relational analysis-based classifier, Comput. Math. Appl. 55 (February (4)) (2008) 680–690.

[7] A. Kampouraki, G. Manis, C. Nikou, Heartbeat time series classification with support vector machines, IEEE Trans. Inform. Technol. Biomed. 13 (July (4)) (2009) 512–518.

[8] R.J. Martis, U.R. Acharya, L.C. Min, ECG beat classification using PCA, LDA, ICA and discrete wavelet transform, Biomed. Signal Process. 8 (September (5)) (2013) 437–448.

[9] W. Jiang, S.G. Kong, Block-based neural networks for personalized ECG signal classification, IEEE Trans. Neural Netw. 18 (November (6)) (2007) 1750–1761.

[10] F. Melgani, Y. Bazi, Classification of electrocardiogram signals with support vector machines and particle swarm optimization, IEEE Trans. Inform. Technol. Biomed. 12 (September (5)) (2008) 667–677.

[11] M. Korürek, B. Dogan, ECG beat classification using particle swarm optimization and radial basis function neural network, Expert Syst. Appl. 37 (December (12)) (2010) 7563–7569.

[12] B. Settles, Active Learning Literature Survey, Univ. of Wisconsin-Madison, Madison, WI, 2010, Technical Report.

[13] E. Pasolli, F. Melgani, Active learning methods for electrocardiographic signal classification, IEEE Trans. Inform. Technol. Biomed. 14 (November (6)) (2010) 1405–1416.

[14] J. Wiens, J.V. Guttag, Active learning applied to patient-adaptive heartbeat classification, in: Proc. NIPS, Vancouver, Canada, 2010, pp. 2442–2450.

[15] Y. Li, F.A. Wessels, D.D. De Ridder, M.J.T. Reinders, Classification in the presence of class noise using a probabilistic kernel Fisher method, Pattern Recognit. 40 (December (12)) (2007) 3349–3357.

[16] D.R. Wilson, Asymptotic properties of nearest rules using edited data, IEEE Trans. Syst. Man Cybern. 2 (July (3)) (1972) 408–421.

[17] L.A. Breslow, D. Aha, Simplifying decision trees: a survey, Knowl. Eng. Rev. 12 (January (1)) (1997) 1–40.

[18] C.E. Brodley, M.A. Friedl, Identifying mislabeled training data, J. Artif. Intell. Res. August (11) (1999) 131–167.

[19] F. Muhlenbach, S. Lallich, D.A. Zighed, Identifying and handling mislabelled instances, J. Intell. Inform. Syst. 22 (January (1)) (2004) 89–109.

[20] U. Rebbapragada, C.E. Brodley, Class noise mitigation through instance weighting, in: Proc. ECML, Warsaw, Poland, 2007, pp. 708–715.

[21] N. Ghoggali, F. Melgani, A genetic automatic ground-truth validation method for multispectral remote sensing images, in: Proc. IGARSS, vol. 4, Boston, MA, 2008, pp. 538–541.

[22] D.E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, Reading, MA, 1989.

[23] L. Davis, Handbook of Genetic Algorithms, Van Nostrand Reinhold, New York, 1991.

[24] N. Ghoggali, F. Melgani, Genetic SVM approach to semisupervised multitemporal classification, IEEE Geosci. Remote Sens. Lett. April (5) (2008) 212–216.

[25] E. Pasolli, F. Melgani, M. Donelli, Automatic analysis of GPR images: a pattern-recognition approach, IEEE Trans. Geosci. Remote Sens. 47 (July (7)) (2009) 2206–2217.

[26] R. Mark, G. Moody, MIT-BIH Arrhythmia Database [Online], 1997, Available on http://ecg. mit.edu/dbinfo.html

[27] V.N. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.

[28] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197.

[29] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, 2nd ed., Wiley, New York, 2001.

[30] Available online. http://www.physionet.org/physiotools/ecgpuwave/src/

[31] J.J. Wei, C.J. Chang, N.K. Shou, G.J. Jan, ECG data compression using truncated singular value decomposition, IEEE Trans. Biomed. Eng. 5 (December (4)) (2001) 290–299.