This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LCSYS.2018.2840427, IEEE Control Systems Letters

A New Classification Algorithm With Guaranteed Sensitivity and Specificity for Medical Applications

Algo Carè, Federico A. Ramponi, Marco C. Campi, *Fellow, IEEE*

***Abstract*—We propose a novel algorithm to construct binary classifiers, in the spirit of the recently proposed Guaranteed Error Machine (GEM) but with a-posteriori assessment of the "support" instances and without the need for a ternary output. We provide rigorous guarantees on the probability of misclassification; differently from GEM, such guarantees aim to bound the conditional probability of error given the true value of the classified instance. The proposed classifier can be tuned in order to give more importance to one of the two kinds of error, and to balance their ratio also in the presence of unbalanced training sets. Guaranteeing the conditional probabilities of error is crucial in many classification problems, in particular medical diagnoses, where being able to push the trade-off between sensitivity (conditional probability of detecting a "true positive") and specificity (conditional probability of detecting a "true negative") towards higher sensitivity is of paramount importance. The application that first motivated our study is the classification of ventricular fibrillation (VF) into cases where restoration of an organized electrical activity is achieved immediately after a defibrillatory shock ("positive"), and cases where prompt resuscitation does not happen ("negative"). We provide experimental evidence that our approach is promising by testing it against three well-known medical datasets, against some data on VF that are available to the authors, and with Monte Carlo simulations.**

***Index Terms*—Pattern recognition and classification; Statistical learning; Healthcare and medical systems**

I. Introduction

MACHINE LEARNING (ML) techniques are gaining popularity as a means for the diagnosis of illnesses and in the prediction of therapy outcomes. In supervised ML a *classifier* is trained from a set of previously recorded pairs $(\mathbf{x}_i, y_i)$, where $\mathbf{x}_i$ are vectors of *features* (parameters extracted from an image, from a blood analysis, etc.) and $y_i \in \{0,1\}$ are labels denoting the corresponding class (for example "healthy" and "ill", or "effective" and "ineffective" in the case of a therapy). When fed with a new vector of features $\mathbf{x}$ the classifier $\hat{y}(\cdot)$ provides an automated prediction $\hat{y}(\mathbf{x})$ for the corresponding y. In this paper we propose a new method to construct a classifier that has guaranteed properties to correctly predict y conditionally to its true value 0 or 1.

The application that originally motivated our study was the first-aid therapy of patients in ventricular fibrillation (VF). European guidelines indicate that VF can be of two kinds. The first one (to which we shall associate the label "1") is promptly reversible with rescue shocks, so that defibrillation results in recovery of circulation and survival, [7]. By contrast, rescue shocks do not result in spontaneous circulation when applied to the other kind of VF ("0"). In this case there is evidence that reperfusion prior to rescue shocks improves defibrillation success and survival, [15]. The quest for effective methods of analysis of VF electrocardiographic (ECG) waveforms and automatic classification to support first-aid decisions (defibrillation or reperfusion followed by defibrillation) is openly endorsed by the last European guidelines, [13]. In a previous work, [1], the Guaranteed Error Machine (GEM) algorithm proposed in [3] was applied to the the problem of VF classification (the features $\mathbf{x}_i \in \mathbb{R}^d$ were extracted by numerical analysis of ECG's) in view of its several attractive properties. First, it allows the user to calibrate the probability of misclassification $\mathsf{P}[\hat{y}(\mathbf{x}) \neq y]$ in a rigorous mathematical way. Second, GEM can lead to good generalization performances even in the presence of many features ($d \gg 1$). On the other hand, the GEM algorithm used in [1] was not able to keep control on the two types of error: misclassifying a "0" (*false positive*) and misclassifying a "1" (*false negative*).

With respect to VF, if a patient of class 1 is classified as 0 then s/he will not be promptly shocked, with potentially fatal consequences. On the other hand, if a patient of class 0 is wrongly classified as 1, the action taken will be to defibrillate as soon as possible, with no benefit. Errors of the first type are *critical* and must be guarded against with the maximum possible care; errors of the second type should be avoided if possible, but given the circumstances are not as critical. In the same way, classifying a patient as healthy when s/he is actually ill has potentially serious consequences and is a critical error, while classifying someone as ill when s/he is not will involve further diagnosis procedures (possibly expensive and invasive), but otherwise cause no harm. Thus, a wise approach to these classification problems is to distinguish between the *conditional* error probabilities $\mathsf{P}[\hat{y}(\mathbf{x}) \neq y \mid y = 1]$ and $\mathsf{P}[\hat{y}(\mathbf{x}) \neq y \mid y = 0]$, and to assign higher importance to the former. In medical statistics it is commonplace to focus on *correct* classification, and to call *sensitivity* the probability $\mathsf{P}[\hat{y}(\mathbf{x}) = y \mid y = 1] = 1 - \mathsf{P}[\hat{y}(\mathbf{x}) \neq y \mid y = 1]$; the probability $\mathsf{P}[\hat{y}(\mathbf{x}) = y \mid y = 0]$ is instead called *specificity*. In the VF classification problem, a 95% sensitivity with a 50% specificity are considered target values, see e.g. [11], [6], [14].

The algorithm introduced in this paper provides a well-principled way for building classifiers in the presence of unbalanced datasets, where different classes are not equally represented, [10], so as to guarantee a desired balance between sensitivity and specificity. Hence, the key novelty of our algorithm is that, while providing a rigorous mathematical guarantee in the same spirit of GEM, it does so for *conditional*

Algo Carè, Federico Alessandro Ramponi and Marco Claudio Campi are with the Dipartimento di ingegneria dell'informazione, Università di Brescia, via Branze 38, 25123 Brescia, Italy (emails: {algo.care,federico.ramponi,marco.campi}@unibs.it).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LCSYS.2018.2840427, IEEE Control Systems Letters

error probabilities, and it can be tuned in order to favor sensitivity over specificity. Of course, how high sensitivity and thereby specificity actually are depends on the number of observations and on how simple the classifier happens to be (many observations and simple constructions lead to stronger claims on sensitivity and specificity). The fundamental point is that, in all cases, the values of sensitivity and specificity are assessable by means of a precise theory prior to using the classifier.

II. The proposed classification algorithm

A training sequence $\mathscr{T} = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N))$, with $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{0,1\}$ has been observed. Assume that $(\mathbf{x}_i, y_i)$, $i = 1, \ldots, N$ are pairs independent and distributed according to a common distribution P; assume, moreover, that the marginal probability of $\mathbf{x}_i$ admits density (except for this assumption, no knowledge of the underlying distribution of $(\mathbf{x}_i, y_i)$ is required to formulate precise guarantees on sensitivity and specificity). Let N_0 be the number of points whose label is zero, $y_i = 0$, and N_1 be the number of points whose label is one, $y_i = 1$ (thus $N_0 + N_1 = N$).

The algorithm below needs to be initialized with a special pair $(\mathbf{x}^*, y^*)$. This can either be one more data point besides $\mathscr{T}$, or more simply an arbitrary point $\mathbf{x}^* \in \mathbb{R}^d$ and an arbitrary label, e.g. $y^* = 0$. The theory developed in the sequel applies to both cases. Moreover, in the algorithm c_0, c_1 are two positive integers called the "complexity" parameters. Their values are chosen by the user and impact on the sensitivity and specificity guarantees as later described in Theorem 2.1.

The classifier is built according to the following algorithm:

Algorithm A

1) Initialize: $\mathbf{x}_c \leftarrow \mathbf{x}^*$ (current center); $y \leftarrow y^*$ (current label); $\mathscr{B} \leftarrow ()$ (list of balls and associated labels); $\mathscr{S}_0 \leftarrow \varnothing$ (set of "support" points $\mathbf{x}_i$ with label 0); $\mathscr{S}_1 \leftarrow \varnothing$ (set of "support" points $\mathbf{x}_i$ with label 1); $\mathscr{R} \leftarrow (\mathbf{x}_1, \ldots, \mathbf{x}_N)$ (list of remaining points).
2) Repeat the following steps,
   a) if there are less than c_{1-y} points in $\mathscr{R}$ with label $1-y$, set $B \leftarrow \mathbb{R}^d$; otherwise let B be the largest open ball, centered at $\mathbf{x}_c$, containing less than c_{1-y} points in $\mathscr{R}$ with label $1-y$. If $B \neq \mathbb{R}^d$, then *almost surely only one* point (with label $1-y$) lies on the boundary of B: denote this point $\hat{\mathbf{x}}$;
   b) add to $\mathscr{S}_{1-y}$ the points in $\mathscr{R}$ with label $1-y$ that belong to B and, if $B \neq \mathbb{R}^d$, also add to $\mathscr{S}_{1-y}$ the boundary point $\hat{\mathbf{x}}$;
   c) remove from $\mathscr{R}$ all points belonging to B;
   d) append (B, y) to the list $\mathscr{B}$;
   e) if $B \neq \mathbb{R}^d$, set $\mathbf{x}_c \leftarrow \hat{\mathbf{x}}$ and $y \leftarrow 1-y$,

   until $B = \mathbb{R}^d$;
3) Define the classifier $\hat{y}(\cdot)$ as follows: for any $\mathbf{x} \in \mathbb{R}^d$, $\hat{y}(\mathbf{x})$:= the label y_i associated with the first ball B_i of the list $\mathscr{B}$ that contains $\mathbf{x}$;
4) Output the classifier $\hat{y}(\cdot)$ and the integers $\mathbf{k}_0 = \mathrm{card}(\mathscr{S}_0) + y$, $\mathbf{k}_1 = \mathrm{card}(\mathscr{S}_1) + (1-y)$.

The algorithm stops when $B = \mathbb{R}^d$. Since (when $B \neq \mathbb{R}^d$) at step 2.(c) the list $\mathscr{R}$ gets reduced, the algorithm certainly comes to termination.

Algorithm **A** constructs a sequence of balls, with alternate labels 0, 1, 0, 1, ..., until the space $\mathbb{R}^d$ is completely covered. At the exit of the algorithm, the sets $\mathscr{S}_0$ and $\mathscr{S}_1$ contain the "important points", those that determine the construction. While Algorithm **A** maintains the same spirit of the GEM algorithm first proposed in [3], it departs from it in many respects. First, the construction only involves balls (GEM considered more complex shapes). Second, GEM was a ternary classifier that admitted the output "unknown". This circumstance is inappropriate whenever a decision has necessarily to be made, hence the new classifier always returns an answer 0 or 1. Third, Algorithm **A** contains a fundamental mechanism to unbalance sensitivity and specificity to favor the former. This is the presence of the complexity parameters c_0 and c_1: increasing the value of c_0 generates 1-labelled balls of larger size so providing higher sensitivity (normally, at the cost of a smaller specificity). This mechanism was not present in GEM.

The next theorem, which quantitatively substantiates the evaluation of sensitivity and specificity, is the main theoretical result of the paper.

Theorem 2.1: Fix small confidence parameters $\beta_0, \beta_1 \in (0,1)$.[1] Define $\varepsilon_0(0) = \varepsilon_1(0) = 0$, $\varepsilon_0(N_0+1) = \varepsilon_1(N_1+1) = 1$, and, for $1 \le k_0 \le N_0$ and $1 \le k_1 \le N_1$, let $\varepsilon_0(k_0), \varepsilon_1(k_1) \in (0,1)$ be quantiles such that

$$\int_{\varepsilon_0(k_0)}^1 f_{k_0, N_0}(p)\, dp = \frac{\beta_0}{N_0}, \quad \int_{\varepsilon_1(k_1)}^1 f_{k_1, N_1}(p)\, dp = \frac{\beta_1}{N_1},$$

where f_{k_0,N_0} and f_{k_1,N_1} are the probability density functions of a $\mathrm{Beta}(k_0, N_0 + 1 - k_0)$ and of a $\mathrm{Beta}(k_1, N_1 + 1 - k_1)$ random variable respectively. Then, irrespective of the distribution according to which the pairs $(\mathbf{x}_i, y_i)$, $i = 1, \ldots, N$, have been sampled, if $(\mathbf{x}, y)$ is a new pair independent of all the $(\mathbf{x}_i, y_i)$ and sampled according to the same distribution, the statements

$$\mathsf{P}[\hat{y}(\mathbf{x}) = 1 \mid y = 0] \le \varepsilon_0(\mathbf{k}_0) \tag{1}$$

$$\mathsf{P}[\hat{y}(\mathbf{x}) = 0 \mid y = 1] \le \varepsilon_1(\mathbf{k}_1) \tag{2}$$

(where $\mathbf{k}_0$ and $\mathbf{k}_1$ are output of the Algorithm **A** along with $\hat{y}(\cdot)$) hold true simultaneously with confidence $1 - \beta_0 - \beta_1$, i.e., the probability with which a training sequence $\mathscr{T}$ returns a classifier such that (1) or (2) are not satisfied is no more than $\beta_0 + \beta_1$. □

III. Proof of the main theorem

Given N, the number of data points, the proportion of 0 and 1-labelled points, as given by N_0 and N_1, is random. First we fix N_0 and N_1 to given values and show that, conditionally to the chosen values of N_0 and N_1, the event

[1] β_0 and β_1 are normally very small values like 10^{-3} or 10^{-4}.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LCSYS.2018.2840427, IEEE Control Systems Letters

$E_0 = \{\mathsf{P}[\hat{y}(\mathbf{x}) = 1 \mid y = 0] > \varepsilon_0(\mathbf{k}_0)\}$ has probability at most β_0, i.e.,

$$\mathsf{P}^N[E_0 \mid N_0, N_1] \leq \beta_0. \tag{3}$$

Since this formula holds for any value of N_0 and N_1, it follows that $\mathsf{P}^N[E_0] \leq \beta_0$. The fact that $\mathsf{P}^N[E_1 \mid N_0, N_1] \leq \beta_1$, and hence $\mathsf{P}^N[E_1] \leq \beta_1$, where $E_1 = \{\mathsf{P}[\hat{y}(\mathbf{x}) = 0 \mid y = 1] > \varepsilon_1(\mathbf{k}_1)\}$, is proven similarly. From $\mathsf{P}^N[E_0] \leq \beta_0$ and $\mathsf{P}^N[E_1] \leq \beta_1$, we then have that (1) and (2) in the statement of Theorem 2.1 hold true simultaneously with probability $1 - \beta_0 - \beta_1$.[2]

To proceed with the proof of (3), notice that changing the order of the pairs $(\mathbf{x}_i, y_i)$, $i = 1, \ldots, N$, does not change the classifier $\hat{y}(\cdot)$ generated by Algorithm **A**. This yields

$$\begin{aligned}
&\mathsf{P}^N\left[\mathsf{P}\left[\hat{y}(\mathbf{x}) = 1 \mid y = 0\right] > \varepsilon_0(\mathbf{k}_0) \mid N_0, N_1\right] \\
&\quad = \mathsf{P}_1^{N_1} \times \mathsf{P}_0^{N_0}\left[\mathsf{P}_0\left[\hat{y}(\mathbf{x}) = 1\right] > \varepsilon_0(\mathbf{k}_0)\right] \\
&\quad = \int_{(\mathbb{R}^d)^{N_1}} \mathsf{P}_0^{N_0}\left[\mathsf{P}_0\left[\hat{y}(\mathbf{x}) = 1\right] > \varepsilon_0(\mathbf{k}_0)\right] \, d\mathsf{P}_1^{N_1},
\end{aligned} \tag{4}$$

where P_0 and P_1 are the conditional distributions of $\mathbf{x}$, given $y = 0$ and given $y = 1$ respectively, and the second equality is Fubini's theorem.

In (4), the integrand $\mathsf{P}_0^{N_0}\left[\mathsf{P}_0\left[\hat{y}(\mathbf{x}) = 1\right] > \varepsilon_0(\mathbf{k}_0)\right]$ is a function of the N_1 points with label 1. From now on, we will assume that the values of the N_1 points with label 1 are fixed, and we will show that

$$\mathsf{P}_0^{N_0}\left[\mathsf{P}_0\left[\hat{y}(\mathbf{x}) = 1\right] > \varepsilon_0(\mathbf{k}_0)\right] \leq \beta_0 \tag{5}$$

irrespective of their values. Integrating over the values of the points with label 1, as is done in (4), gives (3).

To prove (5), start by noting that $\mathbf{k}_0$ is a random variable ranging from 0 to $N_0 + 1$. Hence,

$$\begin{aligned}
&\mathsf{P}_0^{N_0}\left[\mathsf{P}_0\left[\hat{y}(\mathbf{x}) = 1\right] > \varepsilon_0(\mathbf{k}_0)\right] \\
&= \mathsf{P}_0^{N_0}\left[\mathsf{P}_0\left[\hat{y}(\mathbf{x}) = 1\right] > \varepsilon_0(\mathbf{k}_0) \text{ and } 0 \leq \mathbf{k}_0 \leq N_0 + 1\right].
\end{aligned} \tag{6}$$

The case $\mathbf{k}_0 = 0$ happens only when the initialization label is $y^* = 0$ and there are no "ones" ($N_0 = N$, $N_1 = 0$); in this case $\hat{y}(\mathbf{x}) = 0$ for all $\mathbf{x} \in \mathbb{R}^d$, hence $\mathsf{P}_0[\hat{y}(\mathbf{x}) = 1] = 0$ and the inequality $\mathsf{P}_0[\hat{y}(\mathbf{x}) = 1] > \varepsilon_0(0) = 0$ in (6) is false. For $\mathbf{k}_0 = N_0 + 1$, $\varepsilon(\mathbf{k}_0) = 1$, so that $\mathsf{P}_0[\hat{y}(\mathbf{x}) = 1] > \varepsilon_0(N_0 + 1) = 1$ is also clearly false. To handle the other cases $\mathbf{k}_0 = 1, \ldots, N_0$, we introduce N_0 auxiliary classifiers $\hat{y}_1(\cdot), \ldots, \hat{y}_{N_0}(\cdot)$, which are only used to establish certain theoretical relations (these classifiers are not used in practice). Each classifier $\hat{y}_{k_0}(\cdot)$, $k_0 = 1, \ldots, N_0$, is generated by Algorithm $\mathbf{B}(k_0)$ below. Differently from Algorithm **A**, Algorithm $\mathbf{B}(k_0)$ generates a set $\mathscr{S}_0$ whose cardinality is exactly equal to k_0.

Algorithm B(k_0)

1) Initialize: $\mathbf{x}_c \leftarrow \mathbf{x}^*$; $y \leftarrow y^*$; $\mathscr{B} \leftarrow ()$; $\mathscr{S}_0 \leftarrow \varnothing$; $\mathscr{S}_1 \leftarrow \varnothing$; $\mathscr{R} \leftarrow (\mathbf{x}_1, \ldots, \mathbf{x}_N)$;
2) Repeat the following steps,
   a) let $\tilde{c}_0 = \min\{c_0,\ k_0 - \mathrm{card}(\mathscr{S}_0)\}$ and $\tilde{c}_1 = c_1$; if there are less than $\tilde{c}_{1-y}$ points in $\mathscr{R}$ with label $1-y$, set $B \leftarrow \mathbb{R}^d$; otherwise let B be the largest open ball, centered at $\mathbf{x}_c$, containing less than $\tilde{c}_{1-y}$ points in $\mathscr{R}$ with label $1-y$. If $B \neq \mathbb{R}^d$, then *almost surely only one* point (with label $1-y$) lies on the boundary of B: denote this point $\hat{\mathbf{x}}$;
   b) add to $\mathscr{S}_{1-y}$ the points in $\mathscr{R}$ with label $1-y$ that belong to B and, if $B \neq \mathbb{R}^d$, also add to $\mathscr{S}_{1-y}$ the boundary point $\hat{\mathbf{x}}$;
   c) remove from $\mathscr{R}$ all points belonging to B;
   d) append (B, y) to the list $\mathscr{B}$;
   e) if $B \neq \mathbb{R}^d$, set $\mathbf{x}_c \leftarrow \hat{\mathbf{x}}$ and $y \leftarrow 1 - y$,

   until either $\mathrm{card}(\mathscr{S}_0) = k_0$ or $B = \mathbb{R}^d$;
3) If $\mathrm{card}(\mathscr{S}_0) = k_0$, append $(\mathbb{R}^d, 0)$ to the list $\mathscr{B}$; else ($\mathrm{card}(\mathscr{S}_0) < k_0$), find the smallest closed ball B centered at $\mathbf{x}^*$ containing $k_0 - \mathrm{card}(\mathscr{S}_0)$ points in the set {points in $(\mathbf{x}_1, \ldots, \mathbf{x}_N)$ that have label 0 and are not in $\mathscr{S}_0$}; prepend $(B, 1)$ to the list $\mathscr{B}$;
4) Define the classifier $\hat{y}_{k_0}(\cdot)$ as follows: for any $\mathbf{x} \in \mathbb{R}^d$, $\hat{y}_{k_0}(\mathbf{x})$:= the label y_i associated with the first ball B_i of the list $\mathscr{B}$ that contains $\mathbf{x}$; output $\hat{y}_{k_0}(\cdot)$.

Observe now that, whenever $\mathbf{k}_0$ generated by Algorithm **A** takes a value k_0 in $\{1, \ldots, N_0\}$, the only difference between the classifier $\hat{y}(\cdot)$ generated by Algorithm **A** and the classifier $\hat{y}_{k_0}(\cdot)$ generated by Algorithm $\mathbf{B}(k_0)$ is the ball that Algorithm $\mathbf{B}(k_0)$ might have introduced in the "else" part of step 3). Such a ball is prepended to $\mathscr{B}$, and it affects the constructed classifier in such a way that $\hat{y}_{k_0}(\mathbf{x}) = 1$ may happen for values of $\mathbf{x}$ in the ball for which $\hat{y}(\mathbf{x}) = 0$. Hence, for every training sequence $\mathscr{T}$ such that $1 \leq \mathbf{k}_0 \leq N_0$, it holds that

$$\mathsf{P}_0\left[\hat{y}(\mathbf{x}) = 1\right] \leq \mathsf{P}_0\left[\hat{y}_{\mathbf{k}_0}(\mathbf{x}) = 1\right]. \tag{7}$$

Thus, (6) can be bounded as follows

$$\begin{aligned}
&\mathsf{P}_0^{N_0}\left[\mathsf{P}_0\left[\hat{y}(\mathbf{x}) = 1\right] > \varepsilon_0(\mathbf{k}_0) \text{ and } 0 \leq \mathbf{k}_0 \leq N_0 + 1\right] \\
&= \mathsf{P}_0^{N_0}\left[\mathsf{P}_0\left[\hat{y}(\mathbf{x}) = 1\right] > \varepsilon_0(\mathbf{k}_0) \text{ and } 1 \leq \mathbf{k}_0 \leq N_0\right] \\
&\leq \mathsf{P}_0^{N_0}\left[\mathsf{P}_0\left[\hat{y}_{\mathbf{k}_0}(\mathbf{x}) = 1\right] > \varepsilon_0(\mathbf{k}_0) \text{ and } 1 \leq \mathbf{k}_0 \leq N_0\right] \\
&\leq \sum_{k_0=1}^{N_0} \mathsf{P}_0^{N_0}\left[\mathsf{P}_0\left[\hat{y}_{k_0}(\mathbf{x}) = 1\right] > \varepsilon_0(k_0)\right].
\end{aligned}$$

The last part of the proof consists in showing that, for any *fixed* k_0, $\mathsf{P}_0\left[\hat{y}_{k_0}(\mathbf{x}) = 1\right]$ has a $\mathrm{Beta}(k_0, N_0 + 1 - k_0)$ distribution,[3] so that, by the definition of $\varepsilon_0(\cdot)$, we can conclude that

$$\mathsf{P}_0^{N_0}\left[\mathsf{P}_0\left[\hat{y}_{k_0}(\mathbf{x}) = 1\right] > \varepsilon_0(k_0)\right] \leq \frac{\beta_0}{N_0},$$

and statement (5) follows from (6).

Algorithm $\mathbf{B}(k_0)$ is defined for a sequence $\mathscr{T}$ of N data points. As already explained after equation (4), we consider the N_1 points with label 1 fixed and let the N_0 points with label 0 be random and distributed according to P_0.

[2] In view of this approach, we notice that the result of the theorem holds, more strongly, for any values taken by N_0 and N_1, even though in the theorem's statement we have preferred, for the sake of simplicity, not to provide a conditional statement on N_0 and N_1.

[3] In this last part of the proof, we will use a moment identity and a combinatorial argument as in [3], [4].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LCSYS.2018.2840427, IEEE Control Systems Letters

It turns out that investigating the probability of the event $\{\mathsf{P}_0\left[\hat{y}_{k_0}(\mathbf{x})=1\right] > \varepsilon_0(k_0)\}$ requires to consider Algorithm $\mathbf{B}(k_0)$ fed with an enlarged training sequence, which we call $\mathscr{T}_j$, obtained by joining the original sequence of N pairs with more pairs $(\mathbf{x}_{N+1},0),\ldots,(\mathbf{x}_{N+j},0)$ extracted according to P_0, where j is any natural number. Let us maintain the notation $\hat{y}_{k_0}(\cdot)$ for the classifier obtained by feeding $\mathbf{B}(k_0)$ with $\mathscr{T}$ and let us denote $\hat{y}^{(j)}_{k_0}(\cdot)$ the classifier obtained from $\mathscr{T}_j$.

Except for an event with probability zero (a subset of the event {at least one among $\mathbf{x}_{N+1},\ldots,\mathbf{x}_{N+j}$ belongs to the boundary of a ball in the list $\mathscr{B}$ that defines $\hat{y}_{k_0}(\cdot)$}, which has probability zero), the following implications hold:

1) if all the points $\mathbf{x}_{N+1},\ldots,\mathbf{x}_{N+j}$ are classified correctly by $\hat{y}_{k_0}(\cdot)$, then they do not play any role in the construction performed by $\mathbf{B}(k_0)$ when this algorithm is fed with $\mathscr{T}_j$, i.e., $\hat{y}^{(j)}_{k_0}(\cdot)=\hat{y}_{k_0}(\cdot)$, and the k_0 support points ending up in $\mathscr{S}_0$ during the construction of $\hat{y}^{(j)}_{k_0}(\cdot)$ are the same as those in the construction of $\hat{y}_{k_0}(\cdot)$ and belong to $(\mathbf{x}_1,\ldots,\mathbf{x}_N)$;
2) conversely, if at least one point among $\mathbf{x}_{N+1},\ldots,\mathbf{x}_{N+j}$ is misclassified by $\hat{y}_{k_0}(\cdot)$, then at least one among $\mathbf{x}_{N+1},\ldots,\mathbf{x}_{N+j}$ must end up in $\mathscr{S}_0$ during the construction of $\hat{y}^{(j)}_{k_0}(\cdot)$.

Together, these two implications lead to the following fact: almost surely, $\hat{y}_{k_0}(\mathbf{x}_{N+i})=0$ for all $i=1,\ldots,j$ if and only if the support points in $\mathscr{S}_0$ obtained in the construction of $\hat{y}^{(j)}_{k_0}(\cdot)$ are taken from $(\mathbf{x}_1,\ldots,\mathbf{x}_N)$.
Using this fact, we can compute the probability p_j of the event

$$\left\{\hat{y}_{k_0}(\mathbf{x}_{N_0+i})=0 \text{ for all } i=1,\ldots,j\right\}.$$

Start by noticing that, since Algorithm $\mathbf{B}(k_0)$ is permutation invariant, we can assume that the points with label 1 are the first N_1 and those with label 0 are those in the positions N_1+1 through N (when using $\mathscr{T}$) or through $N+j$ (when using $\mathscr{T}_j$). Hence, p_j is the probability that the k_0 support points in the construction of $\hat{y}^{(j)}_{k_0}(\cdot)$, involving N_0+j points with label 0, are taken from the first N_0 points in the positions N_1+1 through N. Exploiting the independence of points and applying simple combinatorics, we then find $p_j=\binom{N_0}{k_0}/\binom{N_0+j}{k_0}$.
On the other hand, applying Fubini's theorem, we get

$$\begin{aligned}
p_j &= \mathsf{P}_0^{N_0+j}\left[\hat{y}_{k_0}(\mathbf{x}_{N+i})=0 \text{ for all } i=1,\ldots,j\right]\\
&= \int_{(\mathbb{R}^d)^{N_0}} \mathsf{P}_0^{j}\left[\hat{y}_{k_0}(\mathbf{x}_{N+i})=0 \text{ for all } i=1,\ldots,j\right]\ d\mathsf{P}_0^{N_0}\\
&= \int_{(\mathbb{R}^d)^{N_0}} \left(\mathsf{P}_0\left[\hat{y}_{k_0}(\mathbf{x})=0\right]\right)^{j}\ d\mathsf{P}_0^{N_0}\\
&= \mathsf{E}\left[\mathsf{P}_0\left[\hat{y}_{k_0}(\mathbf{x})=0\right]^{j}\right],
\end{aligned}$$

i.e., $p_j=\binom{N_0}{k_0}/\binom{N_0+j}{k_0}$ is the j-th order moment of the random variable $\mathsf{P}_0\left[\hat{y}_{k_0}(\mathbf{x})=0\right]$.
Finally, it is easy to check, using e.g. the recursion in [9, beginning of p. 36], that the j-th order moment of a Beta(N_0+1-k_0,k_0) random variable is indeed equal to $p_j=\binom{N_0}{k_0}/\binom{N_0+j}{k_0}$. But then, since the random variable $\mathsf{P}_0\left[\hat{y}_{k_0}(\mathbf{x})=0\right]$ has compact support and all its moments coincide with those of a Beta(N_0+1-k_0,k_0) random variable, we obtain that the distribution of $\mathsf{P}_0\left[\hat{y}_{k_0}(\mathbf{x})=0\right]$ is a Beta(N_0+1-k_0,k_0) (see the uniqueness statement in [12, ch. 2, sec. 12.9, Corollary 1]). This implies that the distribution of $\mathsf{P}_0\left[\hat{y}_{k_0}(\mathbf{x})=1\right] = 1-\mathsf{P}_0\left[\hat{y}_{k_0}(\mathbf{x})=0\right]$ is a Beta(k_0,N_0+1-k_0), as was to be shown. This concludes the proof of the Theorem. □

IV. Experimental evidence and simulations

In this section, we first illustrate the key theoretical properties of the proposed algorithm on easily reproducible synthetic data (Section IV-A), then we apply the algorithm on a few benchmark medical datasets (Section IV-B). We also apply our methodology to the ventricular fibrillation dataset that motivated our research, and conclude that no significant guarantees can be issued on the sensitivity and specificity of the obtained classifier because data are too scarce ($N_1=15$), but we argue that our algorithm might perform well, with strong guarantees, when more data are considered.

A. Synthetic data

In order to illustrate the validity of the theory, we applied our algorithm to a synthetic problem that can be easily reproduced. The problem is that of predicting the output y of the binary function kstest($[x^{(1)},\ldots,x^{(7)}]$,'Alpha',0.005) in the MATLAB Statistics and Machine Learning Toolbox, when the feature vector $\mathbf{x}=[x^{(1)},\ldots,x^{(7)}]$ is uniformly and independently sampled over $[0,1]^7$. For such a distribution of $\mathbf{x}$, the probability that $y=1$ is about $1/10$ of the probability that $y=0$. We took $N=1100$ and, for the sake of simplicity, we considered datasets all having $N_0=1000$ and $N_1=100$.[4] Three conditions for c_0 and c_1 were tried out, namely $c_1:c_0$ equal to $1:1$, to $1:10$, and to $1:50$. For each of these cases, we generated 100 training sets. For each training set, we constructed a classifier[5] and computed the guaranteed lower bounds on its sensitivity and specificity, with $\beta_0=\beta_1=10^{-3}$, so that, according to Theorem 2.1, the bounds are expected to be satisfied unless the generated training set belongs to a set of small probability 0.2%. For each classifier that we obtained, we also evaluated the true sensitivity and specificity using the knowledge of the data generation mechanism (which is usually not available in non-artificial experiments). The results are reported in Fig. 1 (a), (b), and (c). The true sensitivity-specificity couples are connected to the guaranteed bounds by a line: in all of the cases, lines are green to indicate that the bounds are satisfied by the true values. Note that some conservatism in the bound is a matter of necessity since bound satisfaction is enforced with high confidence 99.8%, and sensitivity and specificity are subject to stochastic fluctuation. In Fig. 1 (d), we have connected the true sensitivity-specificity points with the corresponding values $(1-\frac{\mathbf{k}_1}{N_1},1-\frac{\mathbf{k}_0}{N_0})$. A careful inspection of the algorithm reveals that $(1-\frac{\mathbf{k}_1}{N_1},1-\frac{\mathbf{k}_0}{N_0})$ can be interpreted

[4]Note that the statement of Theorem 2.1 is not conditional on N_0 and N_1. However, the theorem's statement extends to the conditional case, see the footnote 2.

[5]In every Monte Carlo run, one extra 0-labelled point was generated and its value was assigned to x^*. In all the numerical studies in this paper, the initial point x^* is a random 0-labelled point that is not included in the count of N_0.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LCSYS.2018.2840427, IEEE Control Systems Letters

as a Leave-One-Out estimate (LOO), [8], of the sensitivity and the specificity of the classifier. Our experiments show that this estimate provides us with a valuable indication of the performance of the classifier. Nonetheless, in many cases the evaluation provided by $(1-\frac{\mathbf{k}_1}{N_1}, 1-\frac{\mathbf{k}_0}{N_0})$ is optimistic (red lines in the picture) and therefore cannot be used reliably as a lower bound. We here remark that $\frac{\mathbf{k}_1}{N_1+1}$ is the mean of the $\text{Beta}(\mathbf{k}_1, N_1+1-\mathbf{k}_1)$ distribution whose quantile is computed in Theorem 2.1. Thus, the guaranteed bounds are obtained by adding a safety margin to the Leave-One-Out estimate, so as to keep under control the variability of the performance of the classifier over the different training sets. These margins are guaranteed for all distributions by which data are generated and yet they are quite tight owing to the fact that they descend from an universal Beta distribution that does not depend on the original distribution of the data.

B. Medical datasets

We applied the classifier developed in this paper to three well-known medical datasets (BreastW, Haberman, and Pima) that had been previously used for testing the GEM algorithm, see [3] for more details and comparisons with other techniques. The algorithm was applied for different choices of $c_1 : c_0$, which resulted in the values $\mathbf{k}_1$ and $\mathbf{k}_0$, and the corresponding guaranteed sensitivity and specificity values (Sens:Spec), reported in Table I. As an additional remark, we note that the reported confidence $1-\beta_1-\beta_0$ is valid when the couple $c_1 : c_0$ is fixed in advance: when n instances of the algorithm are run for n values of $c_1 : c_0$, the bounds are guaranteed with a (lower) overall confidence $1-n\beta_1-n\beta_0$. Finally, we applied the algorithm to the ventricular fibrillation dataset that was presented in [1], with some additional amplitude and frequency features.[6] The results are given in the upper part of Table II. Note that even when $\frac{\mathbf{k}_1}{N_1} \approx \frac{\mathbf{k}_0}{N_0}$ the guaranteed sensitivity and specificity differ considerably. This is due to the fact that the variability of the critical type of error when only 15 samples are available is large, so that a considerably large margin from the leave-one-out estimate $\frac{\mathbf{k}_1}{N_1}$ is necessary to guarantee the claimed sensitivity with the same confidence as for the claimed specificity. Overall, we must conclude that this dataset results in poor guarantees due to the small number of positive instances, and that we need more data. As a proof of concept, we artificially generated more data by using the Synthetic Minority Oversampling TEchnique[7](SMOTE, [5]). We thus obtained an (artificial) dataset with $N_0 = 2476$ and $N_1 = 240$. Results are in the bottom part of Table II. If these data were real, performances would be guaranteed to be very good, close to the values of 95% sensitivity and 50% specificity that are commonly indicated as *target values* in the literature, [11], [6], [14].

[6]Overall, we considered 19 features: Root Mean Square, Average Segment Amplitude, Mean Amplitude, Wave Amplitude, Maximum Value of the Signal, Minimum Value of the Signal, Peak-To-Trough, Mean Slope, Median Slope, AMSA (sum of the absolute value of the product between the amplitude spectral density and the corresponding frequency), absAMSA (absolute value of the sum of the product between the amplitude spectral density and the corresponding frequency), PSA (like AMSA, but with power spectral density instead of amplitude spectral density), Centroid Frequency (frequency of the "center of mass" of the power spectral density), Centroid Power, Dominant Frequency, Edge Frequency, Spectral Flatness Measure, ([1], [2]). Including some features that might look redundant is intentional. In fact, the dimensionality of the feature vector does not enter Theorem 2.1, and we expect that the algorithm performs implicit feature selection as discussed in [1].

V. Conclusions

In this paper we have introduced a new Guaranteed Error Machine algorithm for binary classification. The algorithm inherits from GEM the capability of accommodating the systematic use of many features and, like GEM, is grounded on a rigorous statistical framework that makes it attractive for critical applications. Differently from GEM, the new algorithm has no reject option (its output is always 0 or 1), and has tunable sensitivity and specificity balancing. Moreover, rigorous certificates on the sensitivity and the specificity of the constructed classifier can be issued based on the training set only (i.e., no independent validation set is required). We applied the method to synthetic and medical datasets. Although we have focused on a simple construction based on covering balls, a whole class of algorithms with certificates on sensitivity and specificity can be designed in line with the scheme proposed in this paper. The discussion of the main design knobs (such as the shape of the regions, the possibility of allowing misclassifications in the training set, etc.) will be the subject of future research. In addition, the proposed algorithm contains some freedom in its initialization: the first ball that is being constructed is centered at an observation which, in a generalized form of the algorithm, can be selected by the user. We plan to exploit this degree of freedom to train different classifiers and then boost the performance of the final algorithm by majority-based decisions.

Acknowledgments

We wish to thank Fabio Baronio, Manuela Baronio, Giovanna Perone, Flavio Bonaspetti, and Alberto Apostoli, for sharing with us data and information on the ventricular fibrillation problem in Section IV-B.
This paper was supported by the H&W 2015 program of the University of Brescia under the project "Classificazione della fibrillazione ventricolare a supporto della decisione terapeutica" (CLAFITE).

References

[1] F. Baronio, M. Baronio, M.C. Campi, A. Carè, S. Garatti, and G. Perone. Ventricular Defibrillation: Classification with G.E.M. and a Roadmap for Future Investigations. In *Proc. of the 56th Conf. on Decision and Control (CDC'17)*, Melbourne, VIC, Australia, 2017.
[2] F. Bonaspetti and F. Baronio. Internal Report (University of Brescia), 2011.
[3] M. C. Campi. Classification with guaranteed probability of error. *Machine Learning*, 80(1):63–84, 2010.

[7]We used the MATLAB function available at https://it.mathworks.com/matlabcentral/fileexchange/38830-smote–synthetic-minority-over-sampling-technique-. The function was applied twice to increase the number of positive samples, and twice to increase the number of negative samples so as to preserve the imbalance ratio.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LCSYS.2018.2840427, IEEE Control Systems Letters

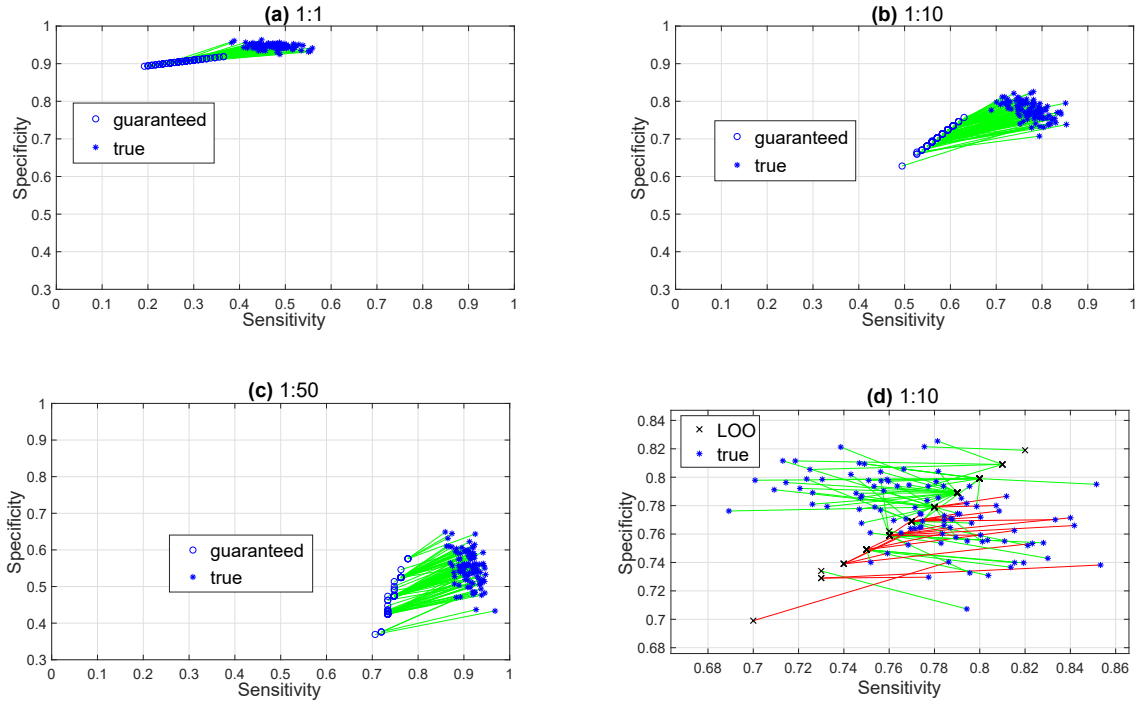

Fig. 1: Results for reproducible artificial data. Labels are generated by using the MATLAB kstest function. $N_0 = 1000$, $N_1 = 100$.

BreastW (239 positive instances, 444 negative instances), $\beta_0 = \beta_1 = 5 \cdot 10^{-3}$							
$c_1 : c_0$	1 : 1	1 : 2	1 : 3	1 : 5	1 : 10	10 : 100	1 : 50
$\mathbf{k}_1 : \mathbf{k}_0$	17 : 17	11 : 21	10 : 28	6 : 28	4 : 33	10 : 17	2 : 66
Sens:*Spec*	83% : 91%	87% : 89%	88% : 87%	90% : 87%	92% : 86%	88% : 91%	94% : 76%

Haberman (75 positive instances, 219 negative instances), $\beta_0 = \beta_1 = 10^{-3}$					
$c_1 : c_0$	1 : 1	1 : 3	1 : 5	1 : 10	1 : 20
$\mathbf{k}_1 : \mathbf{k}_0$	46 : 46	28 : 82	23 : 114	14 : 139	9 : 176
Sens:*Spec*	20% : 67%	41% : 49%	48% : 35%	62% : 24%	71% : 10%

Pima (268 positive instances, 500 negative instances), $\beta_0 = \beta_1 = 10^{-3}$							
$c_1 : c_0$	1 : 1	1 : 2	2 : 4	1 : 4	1 : 8	1 : 10	1 : 50
$\mathbf{k}_1 : \mathbf{k}_0$	125 : 125	88 : 175	96 : 189	61 : 245	38 : 300	33 : 324	9 : 424
Sens:*Spec*	40% : 65%	54% : 55%	51% : 52%	65% : 41%	75% : 30%	70% : 20%	90% : 9%

TABLE I: Guarantees obtained by applying the algorithm with parameters $c_1 : c_0$, β_0, β_1 to benchmark medical datasets [3].

VF dataset, [1] (15 pos, 155 neg), $\beta_0 = \beta_1 = 10^{-2}$			
$c_1 : c_0$	1 : 1	1 : 10	1 : 80
$\mathbf{k}_1$:$\mathbf{k}_0$	9 : 9	5 : 41	2 : 90
Sens:*Spec*	11% : 85%	30% : 59%	51% : 28%

Expanded VF dataset (240 pos, 2477 neg), $\beta_0 = \beta_1 = 10^{-3}$				
$c_1 : c_0$	1 : 1	1 : 10	1 : 80	1 : 240
$\mathbf{k}_1$:$\mathbf{k}_0$	16 : 16	1 : 121	8 : 568	4 : 1055
Sens:*Spec*	84% : 98%	86% : 93%	89% : 73%	92% : 53%

TABLE II

[4] A. Carè, S. Garatti, and M. C. Campi. Scenario min-max optimization and the risk of empirical costs. *SIAM Journal on Optimization*, 25(4):2061–2080, 2015.

[5] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[6] T. Eftestøl, H. Losert, J. Kramer-Johansen, L. Wik, F. Sterz, and P. A. Steen. Independent evaluation of a defibrillation outcome predictor for out-of-hospital cardiac arrested patients. *Resuscitation*, 67(1):55–61, 2005.

[7] M. S. Eisenberg, M. K. Copass, A. P. Hallstrom, B. Blake, L. Bergner, F. A. Short, and L. A. Cobb. Treatment of out-of-hospital cardiac arrests with rapid defibrillation by emergency medical technicians. *New England Journal of Medicine*, 302(25):1379–1383, 1980.

[8] A. Elisseeff, M. Pontil, et al. Leave-one-out error and stability of learning algorithms with applications. In *Advances in Learning Theory: Methods, Models and Applications*, volume 190, pages 111–130. IOS press, 2003.

[9] A. K. Gupta and S. Nadarajah. *Handbook of Beta Distribution and Its Applications*. CRC Press, 2004.

[10] V. López, A. Fernández, S. García, V. Palade, and F. Herrera. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Information Sciences*, 250:113 – 141, 2013.

[11] A. Neurauter, T. Eftestøl, J. Kramer-Johansen, B. S. Abella, K. Sunde, V. Wenzel, K. H. Lindner, J. Eilevstjønn, H. Myklebust, et al. Prediction of countershock success using single features from multiple ventricular fibrillation frequency bands and feature combinations using neural networks. *Resuscitation*, 73(2):253–263, 2007.

[12] A. N. Shiryaev. *Probability*. Springer, New York, NY, USA, 1996.

[13] J. Soar, J. P. Nolan, B. W. Böttiger, G. D. Perkins, C. Lott, P. Carli, et al. European resuscitation council guidelines for resuscitation 2015, Section 3. Adult advanced life support. *Resuscitation*, 95:100–147, 2015.

[14] P. Steen. Is it possible to reliably predict success of defibrillation from the fibrillation waveform? Worksheet ACLS Available at http://circ.ahajournals.org/content/112/22_suppl/b1/tab-figures-data.

[15] L. Wik, T. B. Hansen, F. Fylling, T. Steen, P. Vaagenes, B. H. Auestad, and P. A. Steen. Delaying defibrillation to give basic cardiopulmonary resuscitation to patients with out-of-hospital ventricular fibrillation: a randomized trial. *JAMA*, 289(11):1389–1395, 2003.