2022 IEEE 61st Conference on Decision and Control (CDC)
December 6-9, 2022. Cancún, Mexico

Compression at the service of learning: a case study for the Guaranteed Error Machine

Simone Garatti and Marco C. Campi

2022 IEEE 61st Conference on Decision and Control (CDC) | 978-1-6654-6761-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/CDC51059.2022.9992818

***Abstract*— The scenario approach is a technique for data-driven decision making that has found application in a variety of fields including systems and control design. Although initially conceived in the context of worst-case optimization, the scenario approach has progressively evolved into a general methodology that allows one to keep control on the risk of solutions designed from data according to complex decision processes. In a recent contribution, the theory of compression schemes (a paradigm that plays a fundamental role in statistical learning theory) has been deeply revisited in the wake of the scenario approach, which has led to unprecedentedly sharp generalization and risk quantification results. In this paper, we build on these achievements to gain insight on a classification paradigm called Guaranteed Error Machine (GEM). First, by leveraging the theory of reproducing kernels Hilbert spaces, we introduce a new, more flexible, GEM algorithm, which allows for complex classification geometries. The proposed scheme is then shown to fit into the new compression theory, from which new sharp results for the probability of GEM misclassification are derived in a distribution-free context.**

I. Introduction

We consider supervised classification in which a set of examples (training set) is mapped into a classifier c by a map $\mathcal{A}$. For instance, in supervised binary classification, the classifier is a function from a measured input to a label in $\{-1, 1\}$. Examples are indicated with $z := (x, y)$, where x is the input and y is the label. Given a training set $z_1, \ldots, z_N$, we write $\mathcal{A}(z_1, \ldots, z_N)$ to denote the classifier generated by algorithm $\mathcal{A}$.

Examples are modeled as realizations of random elements over a probability space $(\Omega, \mathcal{F}, \mathbb{P})$; moreover, a list of n examples[1] is always the realization of the first n elements of an independent and identically distributed (i.i.d.) sequence $\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots$.[2] A training set is a list of observed examples. Probability $\mathbb{P}$ models the mechanism by which examples are generated; however, we assume that the user does not know it. The learning algorithms here considered are always permutation invariant, that is, $\mathcal{A}(z_1, \ldots, z_n) = \mathcal{A}(z_{i_1}, \ldots, z_{i_n})$ for any permutation $i_1, \ldots, i_n$ of $1, \ldots, n$.

We use a $\{0, 1\}$-valued function $\ell(c, z)$ to indicate whether or not a classifier c correctly classifies example z: $\ell(c, z) = 0$ signifies correct classification, while $\ell(c, z) = 1$ means incorrect classification. The statistical risk of c is $\mathsf{R}(c) = \mathbb{P}\{\ell(c, \boldsymbol{z}) = 1\}$, where $\boldsymbol{z}$ is a random element distributed as each $\boldsymbol{z}_i$. We aim at evaluating the statistical risk for the classifier $\mathcal{A}(z_1, \ldots, z_N)$ distribution-free, i.e., without any knowledge on $\mathbb{P}$. In this endeavor, we rely on achievements obtained in the so-called scenario approach.

The scenario approach, [1], [2], [3], is nowadays a well-established paradigm for data-driven decision making. It has found wide application in control theory, [4], [5], [6], [7], [8], [9], system identification, [10], [11], [12], [13], [14], and machine learning, [15], [16], [17], [18]. Moreover, many design schemes accommodating diverse design requirements have been introduced within the scenario framework, [19], [20], [21], [22], [23], [24], [25], [26], [27] – see also [28], [29] for general paradigms encompassing most of the existing schemes as special cases.

In this paper, we first review recent results established within the scenario frame for compression schemes, which is a key framework to obtain the sought risk evaluations (Section II). By leveraging this theory, we study the Guaranteed Error Machine (GEM) in Section III. While focused on GEM, this study bears the promise of delivering a new general approach applicable across diverse learning schemes. Section IV closes the paper with a simulation example.

II. A new theory of compression schemes in machine learning

Compression schemes is a framework that has been used to assess the risk of classifiers returned by learning algorithms. The roots of compression schemes can be traced back to the seminal works [30], [31], and the ensuing theory has been deeply investigated in the recent paper [32]. In this section, the results of [32] are briefly summarized since they provide the ground for the analysis of the GEM algorithm introduced in the next section.

Since our learning algorithms return the same classifier independently of the order in which examples are listed, from now on all lists are meant without ordering. So, for example, (a, b, c) is the same as (b, a, c). On the other

S. Garatti is with the Department of Electronics, Information and Bio-engineering, Politecnico di Milano, piazza L. da Vinci 32, 20133 Milano, Italy. Email: simone.garatti@polimi.it.

M.C. Campi is with the Department of Information Engineering, University of Brescia, via Branze 38, 25123 Brescia, Italy. Email: marco.campi@unibs.it.

[1]Note that we use nm which is any integer including 0, while the actual number of data points is N. The reason for having also n is because it plays a role in the analysis.

[2]Throughout, bold symbols, like $\boldsymbol{z}_1$, stand for random elements, while non-bold symbols, like z_1, indicate realizations.

hand, we keep multiplicity, so that (a, a, b) is different from (a, b). In mathematics, a set with repetitions is known under the name of "*multiset*" or "*bag*"; hence, we work with multisets, or bags.

A compression function k is a map from any multiset of examples S to a sub-multiset, that is, $\mathsf{k}(S) \subseteq S$. The following *preference* property is central to our study.

Property 1 (preference): For any couple of multisets of examples S and S' such that $\mathsf{k}(S) \subseteq S' \subseteq S$, it holds that $\mathsf{k}(S) = \mathsf{k}(S')$. ⋆

As is clear, any *preferent* compression function is such that $\mathsf{k}\big(\mathsf{k}(S)\big) = \mathsf{k}(S)$.

Given a learning algorithm $\mathcal{A}$, suppose that there exists a compression function k that ties in with the loss function ℓ according to the following property, which requires that if a new example is misclassified, then adding the new example to the original compressed multiset makes the compressed multiset change.

Property 2 (coherence – part I): For any $n \geq 0$ and any choice of $z_1, \ldots, z_n, z_{n+1} \in \mathcal{Z}$, if $\ell\big(\mathcal{A}(z_1, \ldots, z_n), z_{n+1}\big) = 1$, then $\mathsf{k}\big(\mathsf{k}(z_1, \ldots, z_n), z_{n+1}\big) \neq \mathsf{k}(z_1, \ldots, z_n)$. ⋆

Then, the next Theorem 1 shows that the risk of $\mathcal{A}(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N)$ can be upper bounded by means of the cardinality of the compressed training multiset.

Theorem 1: Given a learning algorithm $\mathcal{A}$, suppose that there exists a compression function k that satisfies the *coherence – part I* Property 2. Assume the *preference* Property 1. For a given $\beta \in (0, 1)$, for $k = 0, 1, \ldots, N-1$ consider the polynomial equation in the t variable

$$\binom{N}{k} t^{N-k} - \frac{\beta}{N} \sum_{i=k}^{N-1} \binom{i}{k} t^{i-k} = 0. \tag{1}$$

For any $k = 0, 1, \ldots, N-1$, equation (1) has exactly one solution in $(0, 1)$, which we denote with t_k. Also define $t_N = 0$.
Let $\epsilon_k := 1 - t_k$. Then, it holds that

$$\mathbb{P}\Big\{\mathsf{R}\big(\mathcal{A}(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N)\big) > \epsilon_{\boldsymbol{k}}\Big\} \leq \beta,$$

where $\boldsymbol{k} = |\mathsf{k}(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N)|$. ⋆

Proof: See [32]. ■

Lower bounds to the risk can be obtained under additional conditions, the *non-associativity* Property 3 and the Property 4 of *non-concentrated* mass, as described in the following.

Property 3 (non-associativity): For any $n \geq 0$ and $p \geq 1$, condition

$$\mathsf{k}(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n, \boldsymbol{z}_{n+i}) = \mathsf{k}(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n),\ i = 1, \ldots, p$$

implies

$$\mathsf{k}(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n, \boldsymbol{z}_{n+1}, \ldots, \boldsymbol{z}_{n+p}) = \mathsf{k}(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n),$$

up to a zero probability event. ⋆

The *non-associativity* property requires that, if the compression does not change adding elements one at a time, then it does not change even when they are added altogether (with the possible exception of an event whose probability is zero).

Property 4 (non-concentrated mass):

$$\mathbb{P}\{\boldsymbol{z} = z\} = 0,\ \forall z \in \mathcal{Z}.$$

⋆

We also strengthen the *coherence* property by adding the following second part.

Property 5 (coherence – part II): For any $n \geq 0$ and $p \geq 1$, condition

$$\mathsf{k}\big(\mathsf{k}(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n), \boldsymbol{z}_{n+1}\big) \neq \mathsf{k}(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n)$$

implies

$$\ell\big(\mathcal{A}(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n), \boldsymbol{z}_{n+1}\big) = 1$$

up to a zero probability event. ⋆

The *coherence – part II* property requires that the change of compression $\mathsf{k}\big(\mathsf{k}(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n), \boldsymbol{z}_{n+1}\big) \neq \mathsf{k}(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n)$ occurs only when the new example is misclassified up to an event of probability zero.

We now have the following result.

Theorem 2: Given a learning algorithm $\mathcal{A}$, suppose that there exists a compression function k that satisfies the *coherence – part I* Property 2 and the *coherence – part II* Property 5. Assume the *preference* Property 1, the *non-associativity* Property 3 and the *non-concentrated mass* Property 4. For a given $\beta \in (0, 1)$, consider for $k = 0, 1, \ldots, N-1$ the polynomial equation in the t variable

$$\binom{N}{k} t^{N-k} - \frac{\beta}{2N} \sum_{i=k}^{N-1} \binom{i}{k} t^{i-k} - \frac{\beta}{6N} \sum_{i=N+1}^{4N} \binom{i}{k} t^{i-k} = 0, \tag{2}$$

and for $k = N$ the polynomial equation

$$1 - \frac{\beta}{6N} \sum_{i=N+1}^{4N} \binom{i}{N} t^{i-N} = 0. \tag{3}$$

For any $k = 0, 1, \ldots, N-1$ equation (2) has exactly two solutions in $[0, +\infty)$, which we denote with $\underline{t}_k$ and $\overline{t}_k$ ($\underline{t}_k \leq \overline{t}_k$). Instead, equation (3) has only one solution in $[0, +\infty)$, which we denote with $\overline{t}_N$, while we define $\underline{t}_N = 0$.
Let $\underline{\epsilon}_k := \max\{0, 1 - \overline{t}_k\}$ and $\overline{\epsilon}_k := 1 - \underline{t}_k$, $k = 0, 1, \ldots, N$. Then, it holds that

$$\mathbb{P}\Big\{\underline{\epsilon}_{\boldsymbol{k}} \leq \mathsf{R}\big(\mathcal{A}(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N)\big) \leq \overline{\epsilon}_{\boldsymbol{k}}\Big\} \geq 1 - \beta,$$

where $\boldsymbol{k} = |\mathsf{k}(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N)|$. ⋆

Proof: See [32]. ■

III. Application to Guaranteed Error Machines (GEM)

The Guaranteed Error Machine (GEM) is a learning algorithm for classification that was first introduced in [15] and then further developed in [18]. GEM returns a ternary-valued classifier, which is also allowed to abstain from classifying in case of doubt. To be specific, letting $z = (x, y)$, with $x \in \mathcal{X}$, a generic set, and $y \in \{-1, 1\}$, a classifier c is a function from $\mathcal{X}$ to $\{-1, 1, 0\}$, where the value 0 is interpreted as admission of being unable to classify. Issuing an incorrect label (-1 in place of 1 or *vice versa*) leads to a mistake, and the theory aims at bounding the probability for this to happen. Correspondingly, the loss function is defined as follows:

$$\ell(c,(x,y)) = \begin{cases} 1, & \text{if } |y - c(x)| = 2 \\ 0, & \text{if } |y - c(x)| = 0 \text{ or } 1. \end{cases}$$

In this paper, we introduce a variant of the GEM algorithm, which takes advantage of the theory of Reproducing Kernel Hilbert Spaces (RKHS, [33], [34], [35]). To describe the operation of GEM, start by introducing a feature map $\varphi : \mathcal{X} \to \mathcal{H}$, where $\mathcal{H}$ is a Hilbert space with inner product $\langle \cdot, \cdot \rangle$. As is well known, φ, $\mathcal{H}$, and $\langle \cdot, \cdot \rangle$ need not be explicitly given, they can be implicitly defined by means of a kernel $K(x, \tilde{x})$ (this is the so-called kernel trick, see e.g. [33]). Theoretical results in RKHS assure that this always corresponds to allocate a suitable couple $\langle \cdot, \cdot \rangle$ and $\varphi(\cdot)$ so that $K(x, \tilde{x}) = \langle \varphi(x), \varphi(\tilde{x}) \rangle$, provided that the kernel is positive definite. We also assume the existence of an ordering on $\mathcal{X}$ (used later to introduce a tie-break rule).

GEM requires that the user chooses an integer $d \geq 1$, which specifies the maximal cardinality for the compression.[3] In loose terms, GEM operates as follows. It is assumed that one has an additional observation $(\bar{x}, \bar{y})$ (besides the training set $S = (x_1, y_1), \ldots, (x_n, y_n)$) that acts as initial "center". GEM constructs the hyper-sphere in $\mathcal{H}$ around $\varphi(\bar{x})$ which is the largest possible under the condition that the hyper-sphere does not include any $\varphi(x_i)$ with label y_i different from $\bar{y}$. All points inside this hyper-sphere are classified as the label $\bar{y}$, and all examples (x_i, y_i) for which $\varphi(x_i)$ is inside the hyper-sphere are removed from the training set. The example that lies on the boundary of the hyper-sphere (and that has therefore prevented the hyper-sphere from further enlarging) is then appointed as the new center (in case of ties, the tie is broken by using the ordering on $\mathcal{X}$) and the procedure is repeated by constructing another hyper-sphere around the new center. This time, only the region given by the difference between the newly constructed hyper-sphere and the first hyper-sphere (which has been already classified) is classified as the label of the second center. This procedure continues the same way and comes to a stop when either the whole space has been classified or the total number of centers is equal to d, in which case the portion of $\mathcal{X}$ that has not been covered is classified as 0. This leads to the algorithm formally described below.

GEM ALGORITHM $\mathcal{G}$

I. SET $q := 0$, $P := S$, $C = \emptyset$ and $x_C = \bar{x}$, $y_C = \bar{y}$;

II. SET $q := q + 1$ and SOLVE problem[4]

$$\begin{aligned} \max_{r \geq 0} \quad & r \\ \text{subject to:} \quad & \|\varphi(x_i) - \varphi(x_C)\| \geq r, \\ & \text{for all } (x_i, y_i) \in P \text{ such that } y_i \neq y_C. \end{aligned} \tag{4}$$

Let r^* be the optimal solution (note that r^* can possibly be $+\infty$);

III. FORM the region $\mathcal{R}_q := \{x \in \mathcal{X} : \|\varphi(x) - \varphi(x_C)\| < r^*\}$ and LET $\ell_q := y_C$; UPDATE P as follows: if $r^* > 0$, then remove from P all the examples with $x_i \in \mathcal{R}_q$; if instead $r^* = 0$,[5] then remove from P the example (x_C, y_C);

IV. IF $r^* < +\infty$, THEN

IV.a SET $C := (C, (x_{i^*}, y_{i^*}))$, where (x_{i^*}, y_{i^*}) is an example in P such that: a. $\|\varphi(x_{i^*}) - \varphi(x_C)\| = r^*$; b. $y_{i^*} \neq y_C$; c. x_{i^*} is smallest in the ordering of $\mathcal{X}$ among all the examples satisfying a. and b.;

IV.b SET $(x_C, y_C) := (x_{i^*}, y_{i^*})$;

V. IF either $|C| = d$ or $P = \emptyset$ THEN STOP and RETURN ℓ_j, $\mathcal{R}_j$, $j = 1, \ldots, q$ and C;
ELSE, GO TO II.

The GEM classifier is defined as

$$\mathcal{G}(S)(x) = \begin{cases} 0, & \text{if } x \notin \mathcal{R}_j \ \forall j = 1, \ldots, q; \\ \ell_{j^*} & \text{otherwise,} \end{cases}$$

where

$$j^* = \min \{ j \quad \in \{1, \ldots, q\} : \ x \in \mathcal{R}_j \}.$$

The compression function for GEM is $\mathsf{k}_{\mathcal{G}}(S) = C$.

We next establish the *preference* and *coherence – part I* properties, required to apply Theorem 1.

$\diamond$ *Preference.* For two multisets of examples S and S' such that $\mathsf{k}_{\mathcal{G}}(S) \subseteq S' \subseteq S$, it is easy to verify that running STEPS I-V with S' as input returns the same output as when these steps are run with input S. Therefore, $\mathsf{k}_{\mathcal{G}}(S') = C = \mathsf{k}_{\mathcal{G}}(S)$ and the *preference* property is satisfied. $\star$

$\diamond$ *Coherence – part I.* Suppose that $\ell(\mathcal{G}(z_1, \ldots, z_n), z_{n+1}) = 1$. By construction of the GEM algorithm, applying STEPS I-V to $\mathsf{k}_{\mathcal{G}}(z_1, \ldots, z_n)$ returns the same output

[3] Selecting a larger value for d reduces the chance of abstention from classifying; when d is larger than the cardinality of the training set, the set of abstention becomes empty.

[4] According to the kernel trick, $\|\varphi(x_i) - \varphi(x_C)\|^2$ can be computed as $K(x_i, x_i) + K(x_C, x_C) - 2K(x_i, x_C)$.

[5] This only happens if there are examples with different labels whose input is x_C.

as when they are applied to $z_1, \ldots, z_n$. If it was that $\mathsf{k}_{\mathcal{G}}(\mathsf{k}_{\mathcal{G}}(z_1, \ldots, z_n), z_{n+1}) = \mathsf{k}_{\mathcal{G}}(z_1, \ldots, z_n)$, then, for the same reason, we would have $\mathcal{G}(\mathsf{k}_{\mathcal{G}}(z_1, \ldots, z_n), z_{n+1}) = \mathcal{G}(\mathsf{k}_{\mathcal{G}}(z_1, \ldots, z_n)) = \mathcal{G}(z_1, \ldots, z_n)$. Consequently, $\ell(\mathcal{G}(\mathsf{k}_{\mathcal{G}}(z_1, \ldots, z_n), z_{n+1}), z_{n+1}) = \ell(\mathcal{G}(z_1, \ldots, z_n), z_{n+1}) = 1$, which is impossible because, by construction, the GEM classifier never misclassifies the examples in the training set. Thus, it must be $\mathsf{k}_{\mathcal{G}}(\mathsf{k}_{\mathcal{G}}(z_1, \ldots, z_n), z_{n+1}) \neq \mathsf{k}_{\mathcal{G}}(z_1, \ldots, z_n)$, that is, the *coherence – part I* Property holds true. ⋆

Applying Theorem 1 we now have the following result.

Theorem 3: **(Risk of GEM)** For any $\beta \in (0, 1)$, it holds that

$$\mathbb{P}\Big\{\mathsf{R}\big(\mathcal{G}(\boldsymbol{S})\big) > \epsilon_{\boldsymbol{k}}\Big\} \leq \beta,$$

where $\boldsymbol{k} = |\mathsf{k}_{\mathcal{G}}(\boldsymbol{S})|$ and ϵ_k is as in Theorem 1. ⋆

Notice also that $|\mathsf{k}_{\mathcal{G}}(\boldsymbol{S})| \leq d$ holds by construction and, since ϵ_k is an increasing function of k, this implies that the bound $\mathsf{R}\big(\mathcal{G}(\boldsymbol{S})\big) \leq \epsilon_d$ is always correct with high confidence $1 - \beta$.[6]

We now turn to lower bounds, which are established by an application of Theorem 2. We start by showing the validity of the *non-associativity* property.

◇ *Non-associativity.* Consider any training set $S = ((x_1, y_1), \ldots, (x_n, y_n))$ and an additional multi-set of examples $S' = ((x_{n+1}, y_{n+1}), \ldots, (x_{n+p}, y_{n+p}))$. Suppose that $\mathsf{k}_{\mathcal{G}}(S \cup S') \neq \mathsf{k}_{\mathcal{G}}(S)$. For this to be, it is required that at least one of these conditions applies: (i) $\ell(\mathcal{G}(S), (x_{n+i}, y_{n+i})) = 1$ for some $i \in \{1, \ldots, p\}$; or, (ii) one of the (x_{n+i}, y_{n+i}), $i \in \{1, \ldots, p\}$, for which $\ell(\mathcal{G}(S), (x_{n+i}, y_{n+i})) = 0$ lies on the boundary of a $\mathcal{R}_j$ and is lower in order than the example that is chosen as center by the algorithm applied to S. However, take an example (x_{n+i}, y_{n+i}) that satisfies either (i) or (ii); then, that example alone makes the compression change. This proves the *non-associativity* property. ⋆

To move on and prove the *non-concentrated mass* and *coherence – part II* properties, we need a mild assumption on the distribution of examples.

Assumption 1: For any $h \in \mathcal{H}$ and $\gamma \in \mathbb{R}$, it holds that

$$\mathbb{P}\{\|\varphi(\boldsymbol{x}) - h\|^2 = \gamma\} = 0.$$

⋆

◇ *Non-concentrated mass.* This immediately follows from Assumption 1: if $\mathbb{P}\{\boldsymbol{z} = \bar{z}\} \neq 0$ for some $\bar{z} = (\bar{x}, \bar{y})$, then Assumption 1 is violated by the choices $c = \varphi(\bar{x})$ and $\gamma = 0$. ⋆

◇ *Coherence – part II.* In view of Assumption 1, $\boldsymbol{x}_{n+1}$ lies on the boundary of a region $\mathcal{R}_j$ with probability zero. On the other hand, when x_{n+1} is not on the boundary, a change of compression only occurs if (x_{n+1}, y_{n+1}) is misclassified, that is, $\ell(\mathcal{G}(S), (x_{n+1}, y_{n+1})) = 1$. This proves the *coherence – part II* property. ⋆

The following theorem now follows from Theorem 2.

Theorem 4: **(Risk of GEM - bounds from below and from above)** Under Assumption 1, for any $\beta \in (0, 1)$, it holds that

$$\mathbb{P}\Big\{\underline{\epsilon}_{\boldsymbol{k}} \leq \mathsf{R}\big(\mathcal{G}(\boldsymbol{S})\big) \leq \bar{\epsilon}_{\boldsymbol{k}}\Big\} \geq 1 - \beta,$$

where $\boldsymbol{k} = |\mathsf{k}_{\mathcal{G}}(\boldsymbol{S})|$ and $\underline{\epsilon}_k$, $\bar{\epsilon}_k$ are as as in Theorem 2. ⋆

IV. Numerical example

In this section, a toy example is considered with the purpose of illustrating both the flexibility of the new GEM algorithm here introduced as well as the sharpness of the evaluation of the statistical risk provided by Theorems 3 and 4.

We take $\boldsymbol{x}_i$ uniformly distributed in the square $[-5, 5] \times [-5, 5]$ and $\boldsymbol{y}_i$ equal to 1 if $\boldsymbol{x}_i$ lies in the circle centered in the origin with radius 3; $\boldsymbol{y}_i = -1$, otherwise. The training set is formed by $N = 500$ examples and the GEM algorithm is run with $d = 50$ and

a. $K(x, \tilde{x}) = x^T \tilde{x}$, which corresponds to $\varphi(x) = x$, i.e. the balls constructed by the GEM algorithm are in fact balls in the original 2D space of x variables;
b. $K(x, \tilde{x}) = x^T \tilde{x} + \|x\|^2 \|\tilde{x}\|^2$, which lifts the x variable into a 3D feature space, enhancing more separability by means of balls (the chosen kernel corresponds to $\varphi(x) = [x^T \ \|x\|^2]^T$).

This is the same example as in Wikipedia – see *https://en.wikipedia.org/wiki/Kernel_method.*

Figure 1 depicts an instance of the training set (crosses mean label equal to 1, while dots correspond to -1). The

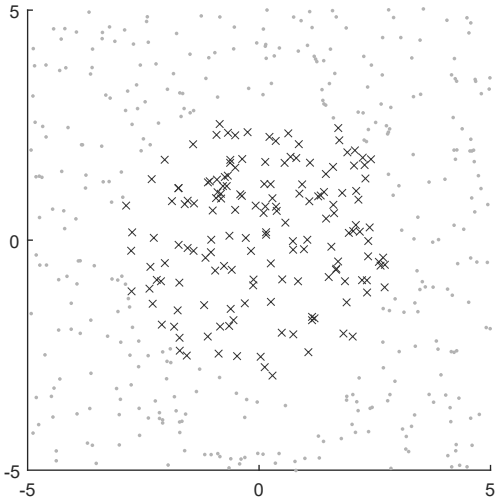

Fig. 1. The dataset in one experiment (crosses = 1, dots = −1).

[6] A similar result would not be possible without resorting to ternary classifiers.

GEM algorithm with these observations as input terminates in a number of iterations smaller than $d = 50$ (and thus it classifies the whole input domain, with no abstention) both for case a and b above. However, in case a, GEM halts after 35 iterations (corresponding to a cardinality of the compressed multiset $\mathsf{k}_{\mathcal{G}}(S)$ equal to 35), while, in case b, GEM only requires 11 iterations (i.e., the cardinality of the compressed set is 11). Intuitively, the more the iterations, the more the fine tuning to the training set (overfitting). This can be appreciated in Figures 2 and 3, where the

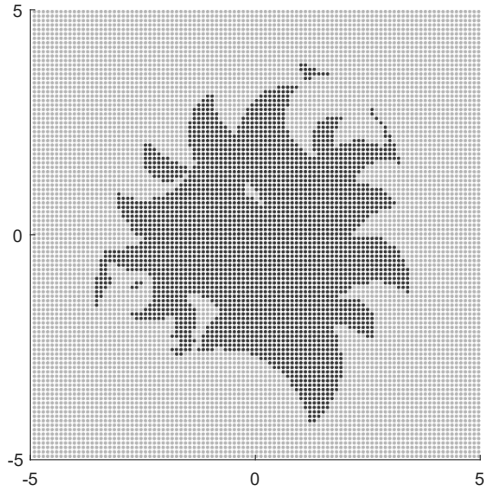

Fig. 2. GEM classifiers for case a. Dark-gray = label 1, light-gray = label -1.

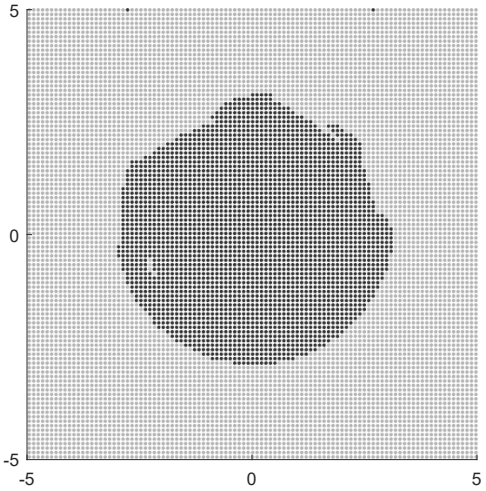

Fig. 3. GEM classifiers for case b. Dark-gray = label 1, light-gray = label -1.

classifiers for case a and b are graphically depicted. Even at an intuitive level, the beneficial effect of lifting the observations into a higher dimension feature space is apparent.

We next apply Theorem 4 to derive rigorous evaluations of the risk (note that assumption 1 is satisfied both in case a and b). Setting $\beta = 10^{-4}$, one gets $\underline{\epsilon}(35) = 2.87\%$ and $\overline{\epsilon}(35) = 13.77\%$ for case a. The true risk[7] associated to the classifier is 8.48%, which lies in the interval $[2.87\%, 13.77\%]$, as predicted by the theory. Similarly, in case b we have that $\underline{\epsilon}(11) = 0.25\%$ and $\overline{\epsilon}(11) = 6.85\%$, while the true risk is 1.72% – once again, the risk is in the interval $[0.25\%, 6.85\%]$.

To better test the validity of Theorem 4, the GEM algorithm is run 400 times in a Monte Carlo simulation, each time drawing a new training set of 500 examples. In the first 200 trials, the kernel $K(x, \tilde{x}) = x^T\tilde{x}$ (case a) is used, while the remaining 200 trials are executed with $K(x, \tilde{x}) = x^T\tilde{x} + \|x\|^2\|\tilde{x}\|^2$ (case b). In each trial, the true risk associated to the classifier is recorded along with the cardinality of the compressed multiset. The pairs

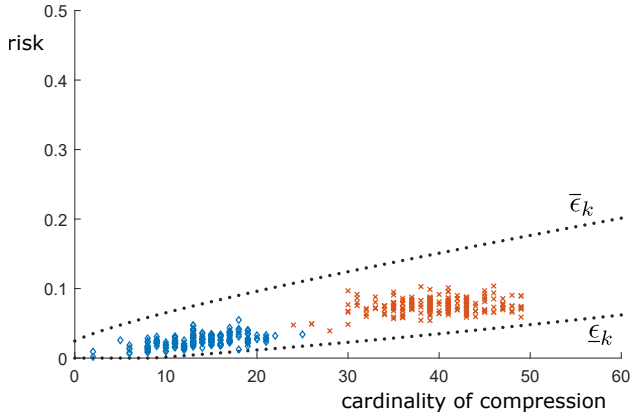

Fig. 4. Stastical risk of GEM classifiers vs. cardinality of the compressed set for $K(x_1, x_2) = x_1^T x_2$ (red crosses) and $K(x_1, x_2) = x_1^T x_2 + \|x_1\|^2\|x_2\|^2$ (blue lozenges). Black dots are $\overline{\epsilon}_k$ and $\underline{\epsilon}_k$ for $\beta = 10^{-4}$.

(cardinality of compression,risk) obtained in the 400 trials are depicted in Figure 4 (red crosses for the first 200 trials with $K(x, \tilde{x}) = x^T\tilde{x}$, blue lozenges for the remaining 200 trials with $K(x, \tilde{x}) = x^T\tilde{x} + \|x\|^2\|\tilde{x}\|^2$). Moreover, the upper and lower limits given by $\underline{\epsilon}(k)$ and $\overline{\epsilon}(k)$ when $\beta = 10^{-4}$ are also displayed (black dots).

Figure 4 once again shows the advantage brought in by using the kernel of case b, which consistently leads to classifiers having lower risks. Moreover, the figure confirms experimentally the validity and the sharpness of Theorem 4: the values for the risk are always within the prescribed intervals (roughly speaking, given that $\beta = 10^{-4}$, Theorem 4 predicts that the risk is not within the interval in at most 1 case out of 10000, which is way bigger than 400 trials here considered). At the same time the spread of the risk values covers well the gap between the lower and upper bounds, showing that these bounds are informative despite they hold distribution free.

[7]The risk can be actually computed because in this simulation example data are artificially generated.

REFERENCES

[1] M. Campi and S. Garatti, *Introduction to Scenario Oprimization*, ser. MOS-SIAM series on Optimization. SIAM, 2018.

[2] G. Calafiore and M. Campi, "Uncertain convex programs: randomized solutions and confidence levels," *Mathematical Programming*, vol. 102, no. 1, pp. 25–46, 2005.

[3] M. Campi and S. Garatti, "The exact feasibility of randomized solutions of uncertain convex programs," *SIAM Journal on Optimization*, vol. 19, no. 3, pp. 1211–1230, 2008.

[4] G. Calafiore and M. Campi, "The scenario approach to robust control design," *IEEE Transactions on Automatic Control*, vol. 51, no. 5, pp. 742–753, 2006.

[5] M. Campi, S. Garatti, and M. Prandini, "The scenario approach for systems and control design," *Annual Reviews in Control*, vol. 33, no. 2, pp. 149 – 157, 2009.

[6] G. Schildbach, L. Fagiano, C. Frei, and M. Morari, "The scenario approach for stochastic model predictive control with bounds on closed-loop constraint violations," *Automatica*, vol. 50, no. 12, pp. 3009–3018, 2014.

[7] S. Grammatico, X. Zhang, K. Margellos, P. Goulart, and J. Lygeros, "A scenario approach for non-convex control design," *IEEE Transactions on Automatic Control*, vol. 61, no. 2, pp. 334–345, 2016.

[8] T. Alamo, R. Tempo, A. Luque, and D. R. Ramirez, "Randomized methods for design of uncertain systems: sample complexity and sequential algorithms," *Automatica*, vol. 51, pp. 160–172, 2015.

[9] M. Campi, S. Garatti, and M. Prandini, "Scenario optimization for mpc," in *Handbook of Model Predictive Control*, S. Raković and W. Levine, Eds. Cham, Switzerland: Birkhäuser, 2019, pp. 445–463.

[10] J. Welsh and C. Rojas, "A scenario based approach to robust experiment design," in *Proceedings of the 15th IFAC Symposium on System Identification*, Saint-Malo, France, 2009.

[11] M. Campi, G. Calafiore, and S. Garatti, "Interval predictor models: identification and reliability," *Automatica*, vol. 45, no. 2, pp. 382–392, 2009.

[12] J. Welsh and H. Kong, "Robust experiment design through randomisation with chance constraints," in *Proceedings of the 18th IFAC World Congress*, Milan, Italy, 2011.

[13] L. Crespo, S. Kenny, and D. Giesy, "Random predictor models for rigorous uncertainty quantification," *International Journal for Uncertainty Quantification*, vol. 5, no. 5, pp. 469–489, 2015.

[14] S. Garatti, M. Campi, and A. Caré, "On a class of interval predictor models with universal reliability," *Automatica*, vol. 110, no. 108542, pp. 1–9, 2019.

[15] M. Campi, "Classification with guaranteed probability of error," *Machine Learning*, vol. 80, pp. 63–84, 2010.

[16] M. Campi and A. Carè, "Random convex programs with l_1-regularization: sparsity and generalization," *SIAM Journal on Control and Optimization*, vol. 51, no. 5, pp. 3532–3557, 2013.

[17] K. Margellos, M. Prandini, and J. Lygeros, "On the connection between compression learning and scenario based single-stage and cascading optimization problems," *IEEE Transactions on Automatic Control*, vol. 60, no. 10, pp. 2716–2721, 2015.

[18] A. Caré, F. Ramponi, and M. Campi, "A new classification algorithm with guaranteed sensitivity and specificity for medical applications," *IEEE Control Systems Letters*, vol. 2, no. 3, pp. 393–398, 2018.

[19] M. Campi and S. Garatti, "A sampling-and-discarding approach to chance-constrained optimization: feasibility and optimality," *Journal of Optimization Theory and Applications*, vol. 148, no. 2, pp. 257–280, 2011.

[20] S. Garatti and M. Campi, "Modulating robustness in control design: principles and algorithms," *IEEE Control Systems*, vol. 33, no. 2, pp. 36–51, 2013.

[21] G. Schildbach, L. Fagiano, and M. Morari, "Randomized solutions to convex programs with multiple chance constraints," *SIAM Journal on Optimization*, vol. 23, no. 4, pp. 2479–2501, 2013.

[22] A. Carè, S. Garatti, and M. Campi, "FAST - Fast Algorithm for the Scenario Technique," *Operations Research*, vol. 62, no. 3, pp. 662–671, 2014.

[23] ——, "Scenario min-max optimization and the risk of empirical costs," *SIAM Journal on Optimization*, vol. 25, no. 4, pp. 2061–2080, 2015.

[24] M. Picallo and F. Dörfler, "Sieving out unnecessary constraints in scenario optimization with an application to power systems," in *Proceedings of the 58th IEEE Conference on Decision and Control (CDC)*, 2019, pp. 6100–6105.

[25] D. Paccagnan and M. Campi, "The scenario approach meets uncertain game theory and variational inequalities," in *Proceeedings of the 58th IEEE Conference on Decision and Control*, 2019.

[26] F. Fele and K. Margellos, "Probably approximately correct nash equilibrium learning," *IEEE Transactions on Automatic Control*, 2020, early access.

[27] L. Romao, K. Margellos, and A. Papachristodoulou, "Tight generalization guarantees for the sampling and discarding approach to scenario optimization," in *Proceedings of the 589th IEEE Conference on Decision and Control (CDC)*, 2020.

[28] M. Campi, S. Garatti, and F. Ramponi, "A general scenario theory for non-convex optimization and decision making," *IEEE Transactions on Automatic Control*, vol. 63, no. 12, pp. 4067–4078, 2018.

[29] S. Garatti and M. Campi, "Risk and complexity in scenario optimization," *Mathematical Programming*, 2019, published on-line. DOI: https://doi.org/10.1007/s10107-019-01446-4.

[30] S. Floyd and M. Warmuth, "Learnability, and the Vapnik-Chervonenkis dimension," *Machine Learning*, vol. 21, pp. 269–304, 1995.

[31] T. Graepel, R. Herbrich, and J.Shawe-Taylor, "PAC-Bayesian compression bounds on the prediction error of learning algorithms for classification," *Machine Learning*, vol. 59, pp. 55–76, 2005.

[32] M. Campi and S. Garatti, "Compression, generalization and learning," *Internal Report*, 2022.

[33] B. Schölkopf and A. Smola, *Learning with kernels*. MIT press, 1998.

[34] A. Lindholm, N. Wahlström, F. Lindsten, and T. Schön, *Machine Learning - A First Course for Engineers and Scientists*. Cambridge University Press, 2022.

[35] O. M. López, , A. M. López, and J. Crossa, *Reproducing Kernel Hilbert Spaces Regression and Classification Methods*. Springer International Publishing, 2022, pp. 251–336.