Learning With Prior Information

M. C. Campi and M. Vidyasagar, *Fellow, IEEE*

***Abstract*—In this paper, a new notion of learnability is introduced, referred to as learnability with prior information (w.p.i.). This notion is weaker than the standard notion of probably approximately correct (PAC) learnability which has been much studied during recent years. A property called "dispersability" is introduced, and it is shown that dispersability plays a key role in the study of learnability w.p.i. Specifically, dispersability of a function class is always a sufficient condition for the function class to be learnable; moreover, in the case of concept classes, dispersability is also a necessary condition for learnability w.p.i. Thus in the case of learnability w.p.i., the dispersability property plays a role similar to the finite metric entropy condition in the case of PAC learnability with a fixed distribution. Next, the notion of learnability w.p.i. is extended to the distribution-free (d.f.) situation, and it is shown that a property called d.f. dispersability (introduced here) is always a sufficient condition for d.f. learnability w.p.i., and is also a necessary condition for d.f. learnability in the case of concept classes. The approach to learning introduced in the present paper is believed to be significant in all problems where a nonlinear system has to be designed based on data. This includes direct inverse control and system identification.**

***Index Terms*—Bayesian learning, dispersability, learning theory, system identification.**

I. Introduction and Problem Definition

A. A Well-Studied Problem: Probably Approximately Correct (PAC) Learnability

CONSIDER a measurable space $(X, \mathcal{X})$ and a probability measure P on $\mathcal{X}$. Further, let F be a set of measurable functions mapping X into $[0, 1]$. Given two functions $f, g \in F$, we can define the distance between them as

$$d_P(f, g) := \int_X |f(x) - g(x)| P(dx).$$

It is easily seen that $d_P(\cdot, \cdot)$ is in fact a pseudometric on F. The quantity $d_P(f, g)$ can be given a very meaningful interpretation. Suppose that a sample point $x \in X$ is selected at random in accordance with the probability P. Suppose also that g is the "true" function—so that $g(x)$ is the "true" outcome—and that f is an approximation to it. Then, $d_P(f, g)$ is the average error we make by using $f(x)$ as an approximation to the true outcome $g(x)$.

A *learning problem* takes place when there is a fixed but unknown function g (usually referred to as the *target function*) and the problem at hand is one of constructing a suitable approximation to g based on observations. In a classical learning setting [26], the observations are collected according to the following scheme: At each instant of time j, an element x_j is drawn at random from X according to P. Moreover, the jth sample is independent of the previous ones. After each sample, an "oracle" returns the value $g(x_j)$ of the unknown target function g evaluated at the randomly generated sample x_j. Thus, at time t, the information available to the learner consists of the *labeled multisample*

$$(x_1, g(x_1)), \ldots, (x_t, g(x_t))$$

where $x_1, \ldots, x_t$ is an i.i.d. sequence distributed according to the probability P. Based on these data, the learner is asked to construct an approximation to the function g. Specifically, he has to choose a function $h_t \in F$ (called the *hypothesis*) with the objective of minimizing the distance $d_P(g, h_t)$. The quantity $d_P(g, h_t)$ is called *generalization error*, since it can be interpreted as the expected error we make by using the hypothesis h_t to predict the value of the *next* outcome $g(x_{t+1})$ of the value of g at a randomly generated element x_{t+1}.

The procedure through which the hypotheses h_ts are generated is called an *algorithm*. Precisely, an algorithm is an indexed family of maps a_t: $(X \times [0, 1])^t \rightarrow F$, $t \geq 1$. Once $\{a_t\}$ is fixed, the hypothesis h_t at time t is obtained simply by applying the algorithm to the available data: $h_t := a_t((x_1, g(x_1)), \ldots, (x_t, g(x_t)))$.

As time increases, the hypothesis h_t is based on an increasing number of data, and is therefore expected to become a progressively better approximation of the target function g. A natural question to ask in connection with the asymptotic behavior of the algorithm is whether h_t tends to g as t tends to infinity. A precise definition of this idea, however, calls for some care as the function h_t is in fact a random element being determined on the basis of the random data $(x_1, g(x_1)), \ldots, (x_t, g(x_t))$. The following definition has by now become standard in learning theory.

Definition 1 (PAC Learnability, [3]): An algorithm $\{a_t\}$ is *probably approximately correct (PAC) to accuracy* ϵ if

$$\lim_{t \to \infty} \sup_{g \in F} P^t\{d_P(g, h_t) > \epsilon\} = 0. \tag{1}$$

The algorithm $\{a_t\}$ is *PAC* if it is PAC to accuracy ϵ, for every $\epsilon > 0$. The function class F is *PAC learnable* if there exists an algorithm that is PAC. □

It should be noted that in Definition 1 probability P is fixed and that algorithm $\{a_t\}$ is required to PAC-learn only for that specific probability. Under the assumption that a function class

Manuscript received September 15, 1999; revised December 10, 2000; revised May 5, 2001 and June 5, 2001. Recommended by Associate Editor T. Parisini. The work of M. Campi was supported by the Centre for AI and Robotics and by MURST under the project "Algorithms and Architectures for the Identification and Control of Industrial Systems."

M. C. Campi is with the Department of Electrical Engineering and Automation, University of Brescia, 25123 Brescia, Italy (e-mail: campi@ing.unibs.it).

M. Vidyasagar was with the Centre for AI and Robotics (CAIR), Bangalore, India. He is now with Tata Consultancy Services, Coromandel House, Secunderabad 500 003, India (e-mail: sagar@atc.tcs.co.in).

Publisher Item Identifier S 0018-9286(01)10341-7.

0018–9286/01$10.00 © 2001 IEEE

F is PAC learnable, an algorithm exists that PAC-learns F. The actual selection of such an algorithm usually requires the explicit knowledge of P. Later on in this section, we will introduce a stronger definition of learnability called distribution-free PAC learnability. There, it is required that a single algorithm is able to learn for all possible probabilities P.

Let us define the quantity

$$r(t, \epsilon;\ F) := \sup_{g \in F} P^t\{d_P(g, h_t) > \epsilon\}.$$

Thus PAC learnability to accuracy ϵ corresponds to the condition that $r(t, \epsilon;\ F)$ approaches zero as $t \to \infty$. Suppose now that an algorithm $\{a_t\}$ PAC learns a given function class F. Suppose an "accuracy parameter" ϵ and a "confidence parameter" δ are specified, and that we are able to determine an integer t_0 with the property that

$$r(t, \epsilon;\ F) \le \delta, \qquad \forall\, t \ge t_0.$$

Then it can be asserted with confidence $1 - \delta$ that, after t_0 samples have been drawn, the generalization error is no larger than ϵ, no matter what the taget function g is. The function $t_0(\delta, \epsilon)$ is referred to as the *sample complexity function*.

A specific, yet significant, instance of learning problem is one in which F is a set of indicator functions on X; that is, $F = \{I_c, c \in C \subset \mathcal{X}\}$. The collection of sets C is called a *concept class* and the problem of estimating I_c is termed a *concept learning problem*.

In Definition 1, a subtle measurability issue arises regarding the definition of the probability $P^t\{d_P(g, h_t) > \epsilon\}$. Since h_t belongs to F, the function $|g - h_t|$ is measurable and $d_P(g, h_t)$ is always well-defined. However, without any extra assumptions on $\{a_t\}$, there is no guarantee that the set $\{d_P(g, h_t) > \epsilon\}$ is measurable, so that $P^t\{d_P(g, h_t) > \epsilon\}$ may not be well-defined. Such an issue is discussed in [6]. In the present paper, here and elsewhere, we gloss over these measurability issues.

Definition 1 can be extended to the case where the probability P is unknown but belongs to some known family of probabilities $\mathcal{P}$. In particular, a case widely studied in the literature is when $\mathcal{P} = \mathcal{P}^*$, the set of *all* probabilities on $(X, \mathcal{X})$, which is known as the *distribution-free learning problem*. In this case, the definition of PAC learnability is as follows.

Definition 2 (Distribution-Free PAC Learnability, [26]): An algorithm $\{a_t\}$ is *distribution-free probably approximately correct (d.f.-PAC) to accuracy* ϵ if

$$\lim_{t \to \infty} \sup_{P \in \mathcal{P}^*} \sup_{g \in F} P^t\{d_P(g, h_t) > \epsilon\} = 0,$$

where $\mathcal{P}^*$ denotes the set of all probabilities on $(X, \mathcal{X})$. The algorithm $\{a_t\}$ is d.f.-PAC if it is d.f.-PAC to accuracy ϵ, for every $\epsilon > 0$. The function class F is d.f.-PAC learnable if there exists an algorithm that is d.f.-PAC. □

Obviously, distribution-free learning is in general a much harder task than learning with a fixed distribution P.

B. Summary of Known Results for PAC Learnability

The classical learning problems described above have been deeply investigated both in the fixed distribution as well as in the distribution-free settings. In particular, for the fixed distribution case, Benedek and Itai [3] have shown that a so-called finite metric entropy condition is both sufficient and necessary for concept learning. Here, we present a general result (Lemma 1) which encompasses both the concept learning case as well as the function learning case.

Given a function family F and a pseudometric d_P on F, a finite collection $\{f_1, \ldots, f_N\} \subseteq F$ is said to be an *ϵ-cover* of F if, for every $g \in F$, there exists an index i such that $d_P(g, f_i) \le \epsilon$. The smallest integer N such that there exists an ϵ-cover of cardinality N is called the *ϵ-covering number* of F with respect to the pseudometric d_P, and is denoted by $N(\epsilon, F, d_P)$. Now the general result is as follows.

Lemma 1: Suppose F is a given function class mapping X into [0, 1], and P is a given probability measure on X. Then, the function class F is PAC learnable if $N(\epsilon, F, d_P) < \infty$, for each $\epsilon > 0$. In case F consists only of binary-valued functions (concept learning), the above condition is also necessary for PAC learnability.

The condition $N(\epsilon, F, d_P) < \infty$ for each $\epsilon > 0$ is referred to as the *finite metric entropy condition*.

In the case of concept learning, the finite metric entropy condition is both necessary and sufficient for PAC learnability. In the case of *function* learning, the finite metric entropy condition is sufficient, but not necessary, for PAC learnability; see [30, Ex. 6.11].

The learnability of a concept class C in a distribution-free setting is closely related to the so-called uniform convergence of empirical probabilities property [28], [29]. This connection was first highlighted in [6] where it was shown that the distribution-free PAC learnability of a concept class is equivalent to the finiteness of its so-called Vapnik–Chervonenkis dimension [27], [28].

Lemma 2 [6]: A concept class is PAC learnable in a distribution-free setting if and only if it has finite VC-dimension.

Extensions of these results to function learning are mainly due to Pollard and Haussler. In particular, in [16], by extending previous results of Pollard, Haussler proves that the finiteness of the Pollard-dimension [24] of a function class F implies that the class is distribution-free PAC learnable. On the other hand, the finiteness of the Pollard-dimension is not required in general for distribution-free PAC learnability to hold. Subsequently, it was shown by Bartlett *et al.* [2] that, in the case of trying to learn a function class under *noisy* measurements, the finiteness of the Pollard-dimension is still not necessary. However, the finiteness of a smaller dimension (the fat-shattering dimension) is necessary and sufficient in that case. We do not dwell on these results here, as they would take us too far afield.

C. Learning With Prior Information

When the conditions that guarantee PAC learnability are not met, a natural question to ask is whether learnability holds in some weaker sense, or else if the class we are dealing with is intrinsically unlearnable for any reasonable definition of learnability. This observation motivates the introduction of other definitions of learnability as alternatives to the classical definition, so that one can "rank" different learning problems in order of difficulty. At present, the study of such alternative learning formulations is still in its infancy.

Benedek and Itai in [4] have proposed the notion of "nonuniform learnability" by simply dropping the uniformity requirement with respect to the target function in Definitions 1 and 2. They give conditions for this property to hold in different situations. An apparent drawback of this approach is that the sample complexity function then explicitly depends upon the target function g [i.e., $t_0(\delta, \epsilon)$ is now replaced by $t_0(g, \delta, \epsilon)$], and consequently the learner does not know when to stop the learning process. This drawback, however, is partially alleviated by a procedure described in [4] based on an alternation of estimation and testing phases. If the learner is content with learning to a prespecified level of accuracy and confidence, this procedure indicates when to stop collecting new data.

A second very interesting stream of literature is the one devoted to the so-called PAC-Bayesian learning, [20], [21], [25]. In PAC-Bayesian learning, the concept class has a prior probability associated with it. This probability is used in order to incorporate *a priori* knowledge in the algorithm. However, differently from the standard Bayesian approach, the results are valid for each single target concept. The basic idea behind this approach is worth mentioning. In standard PAC learning, the accuracy level is fixed and one is asked to find a hypothesis that meets the assigned level of accuracy with a certain confidence. In this context, the accuracy level does not depend on the target concept. In the PAC-Bayesian approach, one drops the uniformity requirement with respect to the target concept and the accuracy level is allowed to depend explicitly on the selected hypothesis. This broadens the applicability of the theory with respect to standard PAC learning. There is however a price one has to pay for such a generality. In PAC-Bayesian, the accuracy is no longer uniform with respect to the target and, therefore, it is not possible to compute *a priori* the size of the data sample that permits one to obtain a specified generalization error. The above mentioned approach has been studied in [20] in connection with countable families of concepts first and then generalized in the same paper to all subsets of arbitrary concept classes. The paper [21] deals with a nontrivial extension of the results in [20] to distributions on arbitrary concept classes.

The present paper is devoted to the study of a new notion of learnability, called here as learnability with prior information (w.p.i.). Instead of insisting on considering each target function as a separate entity, we view the set of target functions as a whole and introduce a probability measure which quantifies the relative importance given to different targets. This formulation of the learning problem is not new and it has been discussed, for example, in [30, Ch. 9]. However, in that reference only the definition of learnability w.p.i. is introduced, and it is shown that any *countable* function class is learnable w.p.i.—a very weak sufficient condition. In the present paper, we present a comprehensive treatment of learnability w.p.i. In particular, we establish necessary and sufficient conditions for learnability w.p.i. of a *concept* class.

Previous work on learning theory in a Bayesian setting—though with a focus completely different from the one in the present paper—can be found in [17], [18]. In these papers, the following problem is addressed: A learner is progressively given the value of the target concept corresponding to randomly extracted x points and is asked to guess the next value. The problem consists in quantifying the probability of error and, in particular, in giving bounds for the cumulative error, i.e., the number of mistakes made in the first m guesses as a function of m. The interesting issue of bounding the probability of error is somewhat complementary to the issue treated in this paper, namely the characterization of function classes which are learnable with prior information.

Before giving a precise definition of learning with prior information, we motivate this approach and highlight some advantages it may have over other learning formulations.

We start with a general observation applicable to any learning problem. Suppose as usual that the problem at hand is one of estimating a target function g based on the observations $(x_1, g(x_1)), \ldots, (x_t, g(x_t))$. Given an algorithm $\{a_t\}$, the accuracy of the estimate h_t computed via $\{a_t\}$, as measured by the number $d_P(g, h_t)$, depends on the extracted multisample $x = (x_1, \ldots, x_t)$ as well as on the target function g. Correspondingly, $d_P(g, h_t)$ may be higher or lower depending on the target g and the multisample x. Both g and x can be regarded as uncertain elements in our learning problem. Therefore, we can claim that the "goodness" of the output of the algorithm is uncertain and depends on two basic uncertain elements: $g \in F$ and $x \in X^t$.

In a classical PAC-learning framework, the two uncertain elements g and x are treated in completely different ways. The learning performance is required to be *uniform* with respect to the one uncertain element $g \in F$, but is permitted to fail with a probability δ with respect to the other uncertain element $x \in X^t$. The requirement of uniformity with respect to g makes the classical learning definition quite demanding, and, not surprisingly, the corresponding learning conditions are not always satisfied.

If one agrees to drop the uniformity requirement with respect to g, then a quite natural approach consists in treating *both* uncertain elements g and x in the same way. This leads to a "fully stochastic" approach which we call *learning with prior information*. This is achieved by equipping the set of target functions F with a probability measure of its own. This probability can describe either an *a priori* available information on the probability of occurrence of the different functions $g \in F$ or, more simply, the relative importance given to different targets. Then all variables involved in the learning process can be viewed as functions defined on the product probability space $F \times X^t$, and all probabilities can be computed in this product probability space.

The approach to learning outlined above may be a meaningful alternative to more standard learning settings in identification and control problems. As an example, consider the problem of estimating a function f by means of a neural network (NN). Typical applications are: direct inverse control problems where the NN is used so as to obtain a dead-beat controller for nonlinear plants [22]; or, more simply, the modeling of a nonlinear dynamical system.

It is well known, see e.g., [1], [7], and [10], that the best size in terms of the number of neurons of the NN results from a compromise between two contrasting needs. On the one hand, the architecture of the NN should be rich enough to have good approximation capabilities. On the other hand, if the size of the NN is too large, it is difficult to train it accurately based on the available finite amount of data points, and this leads to a poor generalization capability. While the first aspect can be captured by a deterministic approximation analysis, the second is better described in a stochastic setting, resulting in a probability, or confidence, that the NN generalizes well.

Clearly, the best compromise between the two sources of errors described above depends on the assumed complexity of the target function f: the more complex f is, the larger the size of the NN should be. A by now standard way of describing the complexity of the target function f is to consider its Fourier transform $\tilde{f}$ and to bound its norm in some way. In Barron [1], use of the following bound is proposed:

$$\int_{R^n} |\omega| \left|\tilde{f}(\omega)\right| d\omega \le C. \tag{2}$$

Once C has been assigned, the size of the NN can be optimized by minimizing the overall estimation error due to the approximation capability of the NN within the function class defined by (2) and the generalization error due to the fact the NN has to be trained based on a finite number of data points.

The above approach relies heavily on the choice of the constant C. If C is chosen to be small, the condition (2) may be too restrictive to include the actual target function f. On the other hand, selecting a large C results in a NN that has many parameters and therefore requires a large number of data points to train. The philosophy of learning with prior information suggests a way around this dilemma. Let $\alpha(C)$ denote an increasing function of C, and suppose that the probability of the target functions satisfying the condition of (2) is $\alpha(C)$. In mathematical terms

$$\Pr\left\{\int_{R^n} |\omega| \left|\tilde{f}(\omega)\right| d\omega \le C\right\} = \alpha(C). \tag{3}$$

Given a desired confidence level δ, one can use (3) to verify whether a specified accuracy can be achieved. A simple way to proceed is to assign a certain fraction of confidence—say $\rho\delta$, $\rho \in [0, 1]$—to the probability that the target function does not belong to $\{\int_{R^n} |\omega||\tilde{f}(\omega)|\, d\omega \le \overline{C}\}$ [that is $\overline{C}$ is selected so that $1 - \alpha(\overline{C}) \le \rho\delta$.] Then, the size of the NN is chosen so as to optimize the generalization error for the class $\{\int_{R^n} |\omega||\tilde{f}(\omega)|\, d\omega \le \overline{C}\}$ with confidence $(1-\rho)\delta$.

The precise formulation of PAC learning with prior information is now introduced. Let $\mathcal{F}$ be a given σ-algebra on F such that the pseudometric d_P is measurable with respect to $\mathcal{F}$, and let Q denote a probability measure on the measurable space $(F, \mathcal{F})$. Throughout, by the symbol Pr_t we indicate the product probability $Q \times P^t$ on the product measurable space $(F \times X^t, \mathcal{F} \otimes \mathcal{X}^t)$.

Definition 3 (PAC Learnability With Prior Information): An algorithm $\{a_t\}$ is *probably approximately correct (PAC) with prior information (w.p.i.) to accuracy* ϵ if

$$\lim_{t\to\infty} \mathrm{Pr}_t\{d_P(g, h_t) > \epsilon\} = 0. \tag{4}$$

The algorithm $\{a_t\}$ is PAC w.p.i. if it is PAC w.p.i. to accuracy ϵ, for every $\epsilon > 0$.

The function class F is PAC learnable w.p.i. if there exists an algorithm that is PAC w.p.i. □

Similarly to Definition 1, in Definition 3 algorithm $\{a_t\}$ is required to learn only for a given fixed probability Pr_t. A function class F is PAC learnable w.p.i. if an algorithm exists able to PAC learn w.p.i. for the given Pr_t. It should be emphasized that this algorithm may be specifically tuned to the given Pr_t. Therefore, selecting the algorithm requires in general explicit knowledge of the probability P and the distribution Q over F.

The learnability condition (4) can be rephrased by saying that PAC learnability w.p.i. holds if the confidence

$$\delta_t := \mathrm{Pr}_t\{d_P(g, h_t) > \epsilon\}$$

approaches zero when $t \to \infty$. In this connection, it is important to remember that δ_t is a probability in $F \times X^t$. Consequently, under the condition $\delta_t \to 0$, no specific statement can be made for *any single* target function g concerning the probability that the generalization error is below the accuracy level ϵ. Rather, δ_t is the *average* probability of success over all the probabilistic elements involved in the learning process.

The fully stochastic approach of Definition 3 offers an important advantage over the nonuniform learning formulation. In the learning problem with prior information one can *a priori* compute a sample complexity function $t_0(\delta, \epsilon)$ such that for $t \ge t_0(\delta, \epsilon)$ learnability holds to accuracy ϵ and confidence δ. This is very similar to the classical PAC learning approach. In contrast, in the nonuniform learning formulation, the sample complexity depends explicitly on the unknown target function, which means that in principle the learner does not know when to stop learning in a specific situation.

The paper is organized as follows. In Section II, we introduce the so-called dispersability property, and present some simple sufficient conditions for a class of functions to be dispersable. Among other results, it is shown that if X is a separable metric space, then *every* family of measurable functions mapping X into $[0, 1]$ is dispersable. In particular, *all* function and concept classes over $\mathbb{R}^n$ are dispersable. In Section III, some algorithms for learning w.p.i. are presented. It turns out that the role of dispersability in PAC learnability w.p.i. is similar to that of finite metric entropy in classical PAC learnability. Specifically, dispersability is a *sufficient* condition for learnability w.p.i. in the case of function classes, and is a *necessary and sufficient* condition for learnability w.p.i. in the case of concept classes; all these claims are established in Section III. Combining the results of Sections II and III shows that *all* function and concept classes over $\mathbb{R}^n$ are learnable w.p.i., for every integer n; this result covers practically all examples in the learning literature. Section IV is devoted to an analysis of the sample complexity of the algorithms for learning w.p.i. introduced in Section III. The problem of PAC learning w.p.i. in a distribution free setting is dealt with in Section V; this problem formulation entails some technicalities. Then, the conditions for learning in this context are established. Finally, Section VI contains the concluding remarks.

II. Dispersable Classes of Functions

In the theory of learning with prior information, a function class F is PAC learnable if there exists an algorithm that is able to return an accurate estimate for *most*, but not necessarily *all*, target functions. In fact, the algorithm is allowed to fail for target functions whose probability Q of occurrence is sufficiently small. In view of this observation, it is expected that a well-posed condition for PAC learnability w.p.i. should simultaneously account for the richness of the function class as well as the probability of occurrence of the different functions in the class.

In this section, we introduce the notion of *dispersability* which is a very natural way of combining the two aspects described above. It turns out that a function class satisfying the finite metric entropy condition is always dispersable. On the other hand, the dispersability condition is much milder than the finite metric entropy condition and holds whenever the "uncoverable" part of F has a small enough probability Q.

The dispersability property applies in particular to concept classes. In Section III, we show that dispersability is a *necessary and sufficient* condition for PAC learning w.p.i. in the case of concept classes.

Consider a partition Π of the function class F, i.e., a collection $\{F_i \in \mathcal{F}\}_{i=1}^r$ such that $\bigcup_{i=1}^r F_i = F$ and $F_i \cap F_j = \emptyset$, $i \neq j$.

Definition 4 (Dispersion Under a Partition): The *dispersion* of the class F under the partition Π is defined as

$$\operatorname{disp}(\Pi) := \sum_{i=1}^{r} \inf_{f\in F} \int_{F_i} d_P(g, f) Q\,(dg). \qquad \square$$

The expression $\inf_{f\in F} \int_{F_i} d_P(g, f)Q\,(dg)$ is a measure of the dispersion of the set F_i (the ith element of the partition Π) where each function $g \in F_i$ is given a weight according to probability Q. Therefore, $\operatorname{disp}(\Pi)$ quantifies the dispersion of a function class once it has been split into the subclasses forming the partition.

Suppose now one is allowed to select a partition Π of given cardinality r so as to minimize the dispersion. The resulting dispersion is the so-called minimal dispersion:

Definition 5 (Minimal Dispersion): The *minimal dispersion* under a partition of cardinality r is defined as

$$\operatorname{disp}(r) := \inf_{\Pi:\, |\Pi|=r} \operatorname{disp}(\Pi). \qquad \square$$

A partition Π is said to be "optimal" when its dispersion is minimal, that is, $\operatorname{disp}(\Pi) = \operatorname{disp}(r)$. Note that an optimal dispersion need not exist in general. However, there will always exist a partition Π of cardinality r such that $\operatorname{disp}(\Pi)$ is arbitrarily close to $\operatorname{disp}(r)$. In the proofs of the various theorems below, it is always assumed that an optimal partition exists. This is purely to reduce notational clutter, and the proofs can be readily amended to cater to the case where optimal partitions do not exist.

Definition 6 (Dispersability): The function class F is *dispersable* if

$$\lim_{r\to\infty} \operatorname{disp}(r) = 0. \qquad \square$$

Thus a function class is dispersable if its dispersion can be made arbitrarily small by considering partitions into more and more subsets.

In the remainder of the section, several results concerning dispersability are proved. First, it is shown that finite metric entropy implies dispersability, and then it is shown that the converse is not true in general.

Lemma 3: Suppose a function class F satisfies the finite metric entropy condition with respect to d_P, and let Q be an arbitrary probability measure on F. Then F is dispersable.

Proof: Recall that a function class F satisfies the finite metric entropy condition if, for every ϵ, no matter how small, the set F can be covered by a finite number of (closed) balls of radius ϵ. Let $N(\epsilon)$ denote the minimum number of balls of radius ϵ needed to cover F. The proof consists of showing that $\operatorname{disp}(r) \leq \epsilon, \forall r \geq N(\epsilon)$, from which it follows that $\lim_{r\to\infty} \operatorname{disp}(r) = 0$, i.e., that F is dispersable. Consider a collection of $N(\epsilon)$ closed balls B_i centered at f_i, for $i = 1, \ldots, N(\epsilon)$, such that $\bigcup_{i=1}^{N(\epsilon)} B_i = F$. Define $F_i = B_i \setminus \bigcup_{j=1}^{i-1} B_j$, $i = 1, \ldots, N(\epsilon)$. Then

$$\begin{aligned}\operatorname{disp}(N(\epsilon)) &= \inf_{\Pi:\, |\Pi|=N(\epsilon)} \operatorname{disp}(\Pi)\\ &\leq \sum_{i=1}^{N(\epsilon)} \inf_{g\in F} \int_{F_i} d_P(g, f)Q\,(dg)\\ &\leq \sum_{i=1}^{N(\epsilon)} \int_{F_i} d_P(g, f_i)Q\,(dg)\\ &\leq \sum_{i=1}^{N(\epsilon)} \epsilon Q(F_i) = \epsilon. \end{aligned} \tag{5}$$

$\square$

On the other hand, dispersability is a milder property than the finite metric entropy property, as shown next.

Lemma 4: Suppose (Y, ρ) is a separable metric space such that $\rho(y, y') \leq 1$, for every $y, y' \in Y$, and let $\mathcal{Y}$ denote the corresponding Borel σ-algebra on Y. Suppose Q is a probability measure on $(Y, \mathcal{Y})$. Then Y is dispersable.

Proof: Given $\epsilon > 0$, select a countable set $\{y_i \in Y\}$ such that, with B_i equal to the closed ball of radius $\epsilon/2$ centered at y_i, we have that $\bigcup_i B_i = Y$. Such a countable set exists since Y is separable. Set $Y_n := \bigcup_{i=1}^{n-1} B_i$, and note that $Q(Y_n) \uparrow 1$. Choose $n(\epsilon)$ such that $Q(Y_{n(\epsilon)}) \geq 1 - \epsilon/2$. Define $F_i := Y_i \backslash Y_{i-1} = B_i \backslash \bigcup_{j=1}^{i-1} B_j$. Then

$$\begin{aligned}\operatorname{disp}(n(\epsilon)) &\leq \sum_{i=1}^{n(\epsilon)-1} \int_{F_i} \rho(y, y_i)Q\,(dy) + Q\left(Y \backslash Y_{n(\epsilon)}\right)\\ &\leq \frac{\epsilon}{2} Q\left(Y_{n(\epsilon)}\right) + Q\left(Y\backslash Y_{n(\epsilon)}\right)\\ &\leq \frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon.\end{aligned}$$

Since ϵ is arbitrary, this implies that Y is dispersable. $\square$

In particular, Lemma 4 implies that every countable set is dispersable under every bounded metric, because a countable set is always separable. On the other hand, it is easy to construct examples of countable sets with a bounded metric that do not satisfy the finite metric entropy condition, which shows that dispersability does not in general imply finite metric entropy.

Example 1: Let $X = [0, 1)$, $\mathcal{X} =$ Borel σ-algebra on X and $P =$ Lebesgue measure on $\mathcal{X}$. Consider the concept class $C = \{c_i\}_{i=1}^{\infty}$, where $c_i = \bigcup_{j=1}^{2^{i-1}} [(2j-1)/2^i, 2j/2^i)$. Clearly, $d_P(c_i, c_j) = 0.5$, whenever $i \neq j$. Thus, $N(\epsilon) = \infty, \forall\, \epsilon < 0.5$, and so this concept class does not satisfy the finite metric entropy condition. However, since it is countable, it follows from Lemma 4 that C is dispersable. □

Next, the mildness of the dispersability condition is highlighted by proving a very general result, namely, that dispersability holds whenever the underlying set X is a separable metric space. In particular, function and concept classes over $\mathbb{R}^n$ are dispersable for every integer n. This result is therefore applicable to practically all examples in the learning literature.

Theorem 1: Suppose that X is a separable metric space and let $\mathcal{X}$ be the associated Borel σ-algebra. Let F denote the family of all measurable functions from X to $[0, 1]$. Let P be any probability measure on $(X, \mathcal{X})$, and let d_P denote the corresponding pseudometric on F. Finally, let Q be any probability measure on the Borel σ-algebra of the metric space (F, d_P). Then, the function class F is dispersable.

Proof: The theorem is proven by showing that (F, d_P) is a separable metric space. Once the separability of (F, d_P) is established, its dispersability follows from Lemma 4.

Note first that the Borel σ-algebra on a separable matric space X is countably generated (by all the balls with rational radius centered on a dense countable subset of X). Thus, $\mathcal{X}$ is countably generated. Next, apply [5, Th. 19.2], which states that the space $L^p(X)$, $1 \leq p < \infty$, where X is a set with σ-finite measure is separable provided that the σ-algebra on X is countably generated. This leads to the conclusion that the space $L^1(X)$ of summable functions on $(X, \mathcal{X})$ is separable. Finally, on observing that (F, d_p) is included in $L^1(X)$, the conclusion follows. □

Even though the notion of dispersability is very general, it is possible to find examples of function classes that are not dispersable.

Example 2: Let $X = R^{[0,1]}$ be the set of real functions defined on the interval $[0, 1]$. The variable $t \in [0, 1]$ is interpreted as time and an element $x \in X$ is a trajectory of a stochastic process. We endow X with a σ-algebra and a probability by means of the standard procedure based on Kolmogorov's existence theorem [5, Th. 36.1]. Specifically, given any finite set of time instants $t_1, \ldots, t_k \in [0, 1]$, define the finite-dimensional distribution corresponding to $t_1, \ldots, t_k$ as the uniform distribution in the hypercube $[0, 1]^k$. This completely defines the probability of cylinder sets, that is sets of form $\{x \in X\colon (x(t_1), \ldots, x(t_k)) \in H\}$, where $x(t_j)$ represents the value of the trajectory x in t_j and H is a Borel set in R^k. Clearly, this system of finite dimensional distributions is consistent, in the sense precisely stated in [5, Sec. 36]. Then, by Kolmogorov's existence theorem, it follows that there exists a probability P defined over the σ-algebra generated by the cylinder sets whose finite dimensional distributions coincide with the given uniform distributions. This completely defines $(X, \mathcal{X}, P)$.

We are interested in concepts of the form $c(t) := \{x \in X\colon x(t) \in [0, 0.5]\}$, where t is some time in $[0, 1]$. Thus the concept $c(t)$ consists of all trajectories x that take on value in the interval $[0, 0.5]$ at time t. Denote by C the corresponding concept class, namely: $C := \{c(t), t \in [0, 1]\}$. It is important to observe that any two distinct concepts $c(t_1)$ and $c(t_2)$ in C are 0.5 apart of each other in the d_P metric. In fact

$$\begin{aligned} &d_P(c(t_1), c(t_2)) \\ &\quad = P(c(t_1)\Delta c(t_2)) \\ &\quad = P(\{x \in X\colon x(t_1) \in [0, 0.5] \text{ and } x(t_2) \in (0.5, 1]\} \\ &\qquad\quad \cup \{x \in X\colon x(t_1) \in (0.5, 1] \text{ and } x(t_2) \in [0, 0.5]\}) \\ &\quad = 0.25 + 0.25 = 0.5. \end{aligned}$$

Clearly, C can be placed in one-to-one correspondence with the interval $[0, 1]$ by the relation $t \leftrightarrow c(t)$. This permits us to immediately introduce a σ-algebra and a probability on C by defining them as the images of the Borel σ-algebra and the Lebesgue measure in $[0, 1]$ through the one-to-one correspondence. The claim is that the concept class C is not dispersable. This can be verified by noting that, for any partition $\Pi = \{C_i\}_{i=1}^r$, we have

$$\begin{aligned} \text{disp}(\Pi) &= \sum_{i=1}^r \inf_{c' \in C} \int_{C_i} d_P(c, c') Q\,(dc) \\ &= \sum_{i=1}^r 0.5 Q(C_i) = 0.5 \end{aligned}$$

that is, under any partition Π, the dispersion is always equal to 0.5. □

III. Connections Between Dispersability and Learnability W.P.I.

A dispersable function class is PAC learnable w.p.i. using a minimum empirical risk algorithm applied to a suitably selected partition of the function class. This is shown in Theorem 3. An analysis of the complexity of this algorithm in the present setting is carried out in Section IV. It turns out that, in the case of *concept* classes, the dispersability condition is also *necessary* for PAC learnability w.p.i. Since this latter result has a very short proof, it is proved first.

In the sequel, we will take the liberty of identifying a set of concepts C with the corresponding function class $F := \{I_c, c \in C\}$. In particular, we shall say that "a concept class C is learnable [dispersable]" if $F = \{I_c, c \in C\}$ is learnable [dispersable].

Theorem 2: A concept class C is PAC learnable w.p.i. only if it is dispersable.

Proof: Consider an algorithm which PAC learns C w.p.i. and denote by $\{h_t\}$ the corresponding random hypothesis sequence. The probability space $(C \times X^t, Q \times P^t)$ in which h_t resides can be embedded in the larger time-invariant probability space $(C \times X^\infty, Q \times P^\infty)$. In this last probability space, the PAC learnability w.p.i. assumption implies that the sequence $\{d_P(c, h_t)\}_{t=1}^\infty$ converges to zero in probability. Therefore, from the sequence $\{d_P(c, h_t)\}_{t=1}^\infty$ it is possible to extract a subsequence $\{d_P(c, h_{t_n})\}_{n=1}^\infty$ that converges to zero $Q \times P^\infty$ almost surely (see, e.g., [14]). This implies that $\forall\, \rho > 0$, $\exists\, C(\rho) \subset C$ such that

1) $Q(C(\rho)) \geq 1 - \rho$;
2) $\lim_{n\to\infty} \sup_{c \in C(\rho)} P^\infty\{d_P(c, h_{t_n}) > \epsilon\} = 0, \forall\, \epsilon > 0$.

By virtue of Theorem 2 in [3], Condition 2 implies that $C(\rho)$ satisfies the finite metric entropy condition, and is therefore dispersable by Lemma 3. Now select a partition Π of $C(\rho)$ such that the dispersion of $C(\rho)$ is less than or equal to ρ; then the partition $\Pi \cup (C \backslash C(\rho))$ of C has a dispersion not greater than 2ρ. Since ρ is arbitrary, this proves that C is dispersable. □

The remainder of the section is devoted to showing that dispersability is a *sufficient* condition for PAC learnability w.p.i. for a function class, by constructing a suitable learning algorithm. We begin by considering a given fixed partition of the function class and introduce a natural procedure for the selection of an element of the partition. Moreover, an estimate of the probability that the corresponding generalization error exceeds a given threshold ϵ is also computed.

Consider a partition $\Pi = \{F_i\}_{i=1}^r$. For the sake of clarity, we assume throughout the sequel that there exist functions f_i, $i = 1, \ldots, r$, minimizing the dispersion of each element $F_i \in \Pi$, i.e., $\int_{F_i} d_P(g, f_i)Q(dg) = \inf_{f \in F} \int_{F_i} d_P(g, f)Q(dg)$. Should this condition not be satisfied, suitable approximations could be used in place of the f_is.

The following procedure is simply a minimal empirical error algorithm for the selection of an f_i in the set $\{f_i\}_{i=1}^r$.

Procedure 1:

1) Determine functions $\{f_i\}_{i=1}^r$ such that $\int_{F_i} d_P(g, f_i) Q(dg) = \inf_{f \in F} \int_{F_i} d_P(g, f)Q(dg)$.
2) Compute the empirical error of each function f_i:
$$\hat{d}_{P,t}(g, f_i) := \frac{1}{t}\sum_{j=1}^{t} |g(x_j) - f_i(x_j)|, \qquad i = 1, \ldots, r.$$
3) Select h_t to be the minimizer of the empirical distance $\hat{d}_{P,t}(g, f_i)$;thus
$$h_t := \operatorname*{argmin}_{f_i,\, i=1,\ldots,r} \hat{d}_{P,t}(g, f_i). \qquad \square$$

A natural question to ask in connection with Procedure 1 is: what is the probability in the product probability space $F \times X^t$ that $d_P(g, h_t)$ exceeds a given value $\epsilon > 0$. The question is dealt with in the following lemma.

Lemma 5: With all notations as above, we have
$$\Pr_t\{d_P(g, h_t) > \epsilon\} \le (r+1)\exp(-t\epsilon^2/8) + \frac{2}{\epsilon}\operatorname{disp}(\Pi), \quad \forall\, \epsilon > 0. \tag{6}$$

Proof: Fix $g \in F$, and choose an $f^0 \in \{f_1, \ldots, f_r\}$ such that
$$d_P(g, f^0) = \min_{1 \le i \le r} d_P(g, f_i).$$

Thus, while h_t is the minimizer of the *empirical* distance between the target function g and the f_i's, f^0 is the minimizer of the *true* distance between g and the f_i's. Note that by definition of $\operatorname{disp}(\Pi)$ we have
$$\int_F d_P(g, f^0)Q(dg) \le \operatorname{disp}(\Pi). \tag{7}$$

We begin by computing the probability that $d_P(g, h_t) - d_P(g, f^0)$ exceeds $\epsilon/2$. Note that if
$$d_P(g, f_i) - \hat{d}_{P,t}(g, f_i) \le \frac{\epsilon}{4} \qquad \text{for } i = 1, \ldots, r, \tag{8}$$
and
$$\hat{d}_{P,t}(g, f^0) - d_P(g, f^0) \le \frac{\epsilon}{4} \tag{9}$$
then it follows that:
$$d_P(g, h_t) - \hat{d}_{P,t}(g, h_t) \le \frac{\epsilon}{4}$$
since $h_t \in \{f_1, \ldots, f_r\}$,
$$\hat{d}_{P,t}(g, h_t) - \hat{d}_{P,t}(g, f^0) \le 0$$
by the manner of choosing h_t,
$$\hat{d}_{P,t}(g, f^0) - d_P(g, f^0) \le \frac{\epsilon}{4}.$$
Adding these three inequalities leads to
$$d_P(g, h_t) - d_P(g, f^0) \le \frac{\epsilon}{2}.$$

Hence, the probability that $d_P(g, h_t) - d_P(g, f^0) > \epsilon/2$ is at most equal to the sum of the probabilities that one of the $r + 1$ inequalities in (8) or (9) is violated. By Hoeffding's inequality (see, e.g., [23]), the probability that any one of these inequalities is violated does not exceed $\exp(-t\epsilon^2/8)$. Hence
$$P^t\{d_P(g, h_t) - d_P(g, f^0) > \epsilon/2\} \le (r+1)\exp(-t\epsilon^2/8). \tag{10}$$
Finally
$$\begin{aligned}
&\Pr_t\{d_P(g, h_t) > \epsilon\} \\
&\quad = \int_F P^t\{d_P(g, h_t) > \epsilon\}Q(dg) \\
&\quad \le \int_F P^t\left(\left\{d_P(g, h_t) - d_P(g, f^0) > \frac{\epsilon}{2}\right\}\right. \\
&\qquad\qquad \left.\cup \left\{d_P(g, f^0) > \frac{\epsilon}{2}\right\}\right) Q(dg) \\
&\quad \le \int_F P^t\left\{d_P(g, h_t) - d_P(g, f^0) > \frac{\epsilon}{2}\right\} Q(dg) \\
&\qquad + \frac{2}{\epsilon}\int_F d_P(g, f^0)Q(dg) \\
&\quad \le (r+1)\exp(-t\epsilon^2/8) + \frac{2}{\epsilon}\operatorname{disp}(\Pi)
\end{aligned}$$
where in the last inequality we have used (10) for bounding the first term and equation (7) for the second one. □

We are now in a position to present our learning algorithm, which consists simply of partitioning the function class so as to reduce the corresponding dispersion to a minimum, and then selecting an hypothesis through Procedure 1. In Theorem 3 below it is shown that this algorithm PAC learns w.p.i. when applied to a dispersable class provided that the rate of growth of the partition size is subexponential.

Algorithm 1: Select an increasing integer-valued function $r(t) \uparrow \infty$. At time t, do the following:

1) determine an optimal partition Π_t of cardinality $r(t)$ [thus, $\operatorname{disp}(\Pi_t) = \operatorname{disp}(r(t))$];
2) select h_t by applying Procedure 1 to the partition Π_t. □

In Algorithm 1 it is assumed that an optimal partition exists; if not, "nearly optimal" partitions can be used instead, and the proof below can be modified appropriately.

Theorem 3: If F is dispersable and $r(t) = \exp(o(t))$, then Algorithm 1 PAC learns class F w.p.i.

Proof: The conclusion follows readily from Lemma 5, which states that

$$\begin{aligned}\Pr_t\{d_P(g, h_t) > \epsilon\} &\le (r(t)+1)\exp(-t\epsilon^2/8) + \frac{2}{\epsilon}\operatorname{disp}(r(t)) \\ &= \exp(o(t) - t\epsilon^2/8) + \frac{2}{\epsilon}\operatorname{disp}(r(t)). \qquad (11)\end{aligned}$$

Since $r(t) \uparrow \infty$, the right side of (11) tends to zero for every $\epsilon > 0$. Hence, the algorithm PAC learns w.p.i. □

The theorems proved thus far permit us to draw some very general conclusions regarding learnability w.p.i.

Theorem 4: A concept class is PAC-learnable w.p.i. if and only if it is dispersable.

Proof: The "only if" part is proven in Theorem 2. The "if" part follows from Theorem 3 which proves the existence of an algorithm that PAC learns class F w.p.i. □

Theorem 5: Let X be a separable metric space, equipped with the associated Borel σ-algebra. Let F denote the set of all measurable functions mapping X into $[0, 1]$. Finally, let Q be any probability measure on F. Then F is PAC learnable w.p.i.

The proof follows readily from Theorems 1 and 3.

Theorem 5 shows that in the most widely studied situation where X is a subset of some Euclidean space $\mathbb{R}^n$ for some integer n, learnability w.p.i. is automatic.

It is interesting to note that in Algorithm 1, attention is first restricted to a finite number of candidate hypotheses (functions $\{f_i\}_{i=1}^r$), and then this number is permitted to go to infinity (in a controlled way) as the number of data points increases. The reason for this is that if too many hypotheses are considered at the same time, the probability that the generalization error for all of them can be correctly estimated from data is very low. Then, selecting a hypothesis which exhibits good adherence to data gives no guarantee that this hypothesis generalizes well.

This idea of restricting attention to a subclass of hypotheses is standard in the statistical literature and it has been used in many different forms and contexts (see, e.g., [15], where the notion of "sieve" is introduced in connection with the problem of estimating probability measures). The very interesting fact within the framework of the present paper is that restricting attention to a finite set of concepts as indicated in Algorithm 1 works *whenever* a concept class is PAC learnable w.p.i.

We now present an alternative to Procedure 1. In the first step of Procedure 1, one is obliged to determine functions f_is that are at a minimal average distance from the functions in the ith element of the partition Π. However, determining these functions may be very difficult. Instead, one possibility is to select an $f_i \in F_i$ at random for each i, according to the probability Q. This leads to the following alternative to Procedure 1.

Procedure 2:

1) For $i = 1, \ldots, r$, select at random a function f_i out of F_i according to probability Q restricted to set F_i;
2) and 3) as in Procedure 1. □

It is natural to ask whether a result similar to Lemma 5 still holds for Procedure 2. This is indeed the case, as shown next. Note that there is now an extra element of randomness in Procedure 2 since at the first step of this procedure functions f_is are randomly selected. As a consequence, the hypothesis h_t is now a random element in the probability space $F \times X^t \times F_1 \times \cdots \times F_r$. Denoting by Q_{F_i} the probability Q restricted to F_i (i.e., $Q_{F_i} = Q/Q(F_i)$), the probability on $F \times X^t \times F_1 \times \cdots \times F_r$ is then given by $\Pr_t := Q \times P^t \times Q_{F_1} \times \cdots \times Q_{F_r}$. The generalization of Lemma 5 to Procedure 2 makes reference to this probability.

Lemma 6: With h_t generated according to Procedure 2 we have

$$\Pr\{d_P(g, h_t) > \epsilon\} \le (r+1)\exp(-t\epsilon^2/8) + \frac{4}{\epsilon}\operatorname{disp}(\Pi), \quad \forall\, \epsilon > 0.$$

By comparing Lemmas 5 and 6, we see that the upper bound for the probability of error with the random Procedure 2 increases by a factor less than 2 over the upper bound of the probability of error with Procedure 1.

Proof: The proof is analogous to that of Lemma 5 and therefore omitted. □

With Lemma 6 in place it is possible to prove a result analogous to Theorem 3 for the following variant of Algorithm 1.

Algorithm 2: Select an increasing integer function $r(t) \uparrow \infty$. At time t, do the following:

1) determine an optimal partition $\Pi_t := \{F_{t,1}, \ldots, F_{t,r(t)}\}$;
2) select h_t by applying Procedure 2 to Π_t. □

Theorem 6: If F is dispersable and $r(t) = \exp(o(t))$, then h_t computed through Algorithm 2 satisfies

$$\lim_{t\to\infty} \Pr_t\{d_P(f, h_t) > \epsilon\} = 0, \qquad \forall\, \epsilon > 0$$

where $\Pr_t := Q \times P^t \times Q_{F_{t,1}} \times \cdots \times Q_{F_{t,r(t)}}$. □

IV. Sample Complexity Evaluation

In this section, we examine the sample complexity of Algorithm 1. The starting point is the bound (6) in Lemma 5, which states that, given a partition Π of cardinality r, the hypothesis h_t generated through Procedure 1 satisfies

$$\Pr_t\{d_P(g, h_t) > \epsilon\} \le (r+1)\exp(-t\epsilon^2/8) + \frac{2}{\epsilon}\operatorname{disp}(\Pi).$$

Note that the above result holds true for every r, t, and ϵ.

When Π is an optimal partition (as is the case in Algorithm 1), we have $\operatorname{disp}(\Pi) = \operatorname{disp}(r)$ and

$$\delta(r, t, \epsilon) := (r+1)\exp(-t\epsilon^2/8) + \frac{2}{\epsilon}\operatorname{disp}(r)$$

is the so-called *confidence function*. This expression for $\delta(r, t, \epsilon)$ can be minimized with respect to the parameter r as a function of t and ϵ. Letting $r_0(t, \epsilon)$ denote the optimal value for r, the minimal confidence δ is defined as

$$\delta_0(t, \epsilon) := \delta(r_0(t, \epsilon), t, \epsilon). \qquad (12)$$

The function $\delta = \delta_0(t, \epsilon)$ can be made explicit with respect to t (with a little caution due to the fact that t is an integer variable) as follows:

$$t_0(\delta, \epsilon) := \text{the smallest integer } t \text{ such that } \delta_0(t, \epsilon) \leq \delta.$$

In the sequel, $t_0(\delta, \epsilon)$ is referred to as the *sample complexity function*.

It is important to give a correct interpretation to the function $t_0(\delta, \epsilon)$. Given an accuracy ϵ and a confidence δ, the integer $t_0(\delta, \epsilon)$ is such that $\delta(r_0(t_0(\delta, \epsilon), \epsilon), t_0(\delta, \epsilon), \epsilon) \leq \delta$ [see (12)]. In other words, the confidence at time $t_0(\delta, \epsilon)$ of the relation $d_P(g, h_t) \leq 1-\epsilon$ is at least $1-\delta$ provided that the partition Π has cardinality $r_0(t_0(\delta, \epsilon), \epsilon)$. On the other hand, in Algorithm 1 the integer function $r(t)$ has to be chosen at the outset and is not allowed to be a function of ϵ at time $t = t_0(\delta, \epsilon)$. So: $r(t_0(\delta, \epsilon)) \neq r_0(t_0(\delta, \epsilon), \epsilon)$ in general.

Clearly, a sensible way to determine function $r(t)$ in Algorithm 1 is to choose first a function $\epsilon(t)$ and then optimize r by selecting $r(t) = r_0(t, \epsilon(t))$. If this is the case, the confidence δ at time t is in fact optimal for accuracy $\epsilon(t)$, i.e., $\delta = \delta_0(t, \epsilon(t))$, but it is in general suboptimal for $\epsilon \neq \epsilon(t)$.

In conclusion, the sample complexity function provides a theoretical lower bound which can only be partially achieved by a given algorithm. Precisely, if $r(t) = r_0(t, \epsilon(t))$ for some prespecified function $\epsilon(\cdot)$ and, for some t, we select $\epsilon = \epsilon(t)$ and $\delta = \delta_0(t, \epsilon(t))$, then we achieve accuracy ϵ with confidence δ at time t and t is in fact optimal: $t = t_0(\delta, \epsilon)$.

In order to determine an explicit expression for the sample complexity function, one has to introduce some specific form for the minimal dispersion function $\operatorname{disp}(\cdot)$. Here, as an example, we examine the case in which

$$\operatorname{disp}(r) = O(1/r^{\alpha}) \tag{13}$$

for some constant $\alpha > 0$.

Example 3: Suppose that F satisfies the finite metric entropy condition and that $N(\epsilon) = O(1/\epsilon^{\lambda})$, for some constant $\lambda > 0$. This is quite a common situation; see for instance several examples reported in [19]. Then, by using (5), it is readily seen that (13) holds in this case with $\alpha = 1/\lambda$. □

Example 4: Consider again Example 1 in Section II and assume that $Q(c_i) = i^{-\lambda}/\sum_{i=1}^{\infty} i^{-\lambda}$, for some $\lambda > 1$. In this case the finite metric entropy condition is violated. However, an easy computation shows that equation (13) is still satisfied with $\alpha = \lambda - 1$. □

The first step in the determination of the sample complexity function is the optimization of the confidence function

$$\delta(r, t, \epsilon) := (r+1)\exp(-t\epsilon^2/8) + \frac{2}{\epsilon} M_1 \frac{1}{r^{\alpha}},$$

where M_1 is a suitable constant. Selecting

$$r(t, \epsilon) = -1 + \left[\frac{1}{\epsilon}\exp(t\epsilon^2/8)\right]^{1/(\alpha+1)}$$

leads to

$$\delta_0(t, \epsilon) \leq \delta(r(t, \epsilon), t, \epsilon) \leq M_2 \left[\frac{1}{\epsilon}\exp(-t\epsilon^2\alpha/8)\right]^{1/(\alpha+1)}$$

where M_2 is a suitable constant. This equation can be easily made explicit with respect to t, leading to the sample complexity function

$$t_0(\delta, \epsilon) = O\left(\frac{1}{\epsilon^2}\left[\log\frac{1}{\epsilon} + \log\frac{1}{\delta}\right]\right). \tag{14}$$

Other bounds where t_0 exhibits a dependence on ϵ of the form $O(1/\epsilon)$ can be achieved by tightening the bound in (6) by using Bernstein's inequality (see, e.g. [11, Ch. 8]).

Note that the sample complexity function (14) is similar to the one derived in a classical concept learning context in [3]. The relation (14), however, has a different interpretation from the results in [3] at least in two respects. First, the confidence δ is computed here as a probability in the product probability space $F \times X^t$ rather than a probability in the sample space X^t. Secondly, results in [3] are worked out under the finite metric entropy condition whereas the present results make use of the milder dispersability condition.

V. Distribution-Free Learning With Prior Information

A. Mathematical Setting and Definitions

This section is devoted to the problem of learning with prior information in the case in which the probability P is not fixed and it can in fact be *any* probability on $\mathcal{X}$. Define $\mathcal{P}^*$ to be the set of all probabilities on $\mathcal{X}$.

Let $\mathcal{F}$ denote a given σ-algebra on F, and let Q denote a probability measure on $(F, \mathcal{F})$. The probability Q constitutes the *a priori* probability that a function f happens to be the target function, or else the relative importance placed on different target functions. The probability Q is known to the learner. According to the philosophy of learning with an arbitrary distribution, given a function $g \in F$, the probability $P(g)$ according to which the samples x_j are collected is allowed to be any probability in $\mathcal{P}^*$. Moreover, the probability P may be different for different functions g. By the symbol K we denote a kernel of probabilities indexed by $g \in F$

$$K := \{P(g), g \in F\}$$

that is, for a given g, $P(g)$ is a probability over X and the probability $P(g, A)$ of a set $A \in \mathcal{X}$ is $\mathcal{F}$-measurable. In the context of distribution-free learning, K plays a role similar to that of P in the fixed distribution setting. Throughout, it is assumed that K is not known and can be any kernel. The set of all kernels is denoted by $\mathcal{K}^*$.

Given a kernel K, the probability Q allows us to define a corresponding probability $\Pr_t$ in the product measurable space $(F \times X^t, \mathcal{F} \otimes \mathcal{X}^t)$ as the unique probability measure which extends the definition $\Pr_t(A \times B) := \int_B P^t(g, A)Q\,(dg)$, $A \in \mathcal{X}^t$, $B \in \mathcal{F}$, to the σ-algebra $\mathcal{F} \otimes \mathcal{X}^t$.

Our first step in the development of a distribution-free learning theory with prior information is the extension of the definitions in Sections I and II to the present setting.

Definition 7 (Distribution-Free PAC Learnability With Prior Information): An algorithm $\{a_t\}$ is *d.f. PAC w.p.i. to accuracy* ϵ if

$$\lim_{t\to\infty} \sup_{K \in \mathcal{K}^*} \Pr_t\left\{d_{P(g)}(g, h_t) > \epsilon\right\} = 0. \tag{15}$$

The algorithm $\{a_t\}$ is d.f. PAC w.p.i. if it is d.f. PAC w.p.i. to accuracy ϵ, for every $\epsilon > 0$. The function class F is *d.f. PAC learnable w.p.i.* if there exists an algorithm that is d.f. PAC w.p.i. □

The distinctive feature of Definition 7 as compared to Definition 3 is that in (15) convergence is required to hold uniformly in $K \in \mathcal{K}^*$; that is, the probability P is allowed to depend on g and this dependence can be arbitrary since $\{P(g)\}$ can be any kernel. Clearly, the convergence requirement in (15) is stronger than the one in (4).

Next we wish to extend the notion of dispersability to the distribution-free setting. For this purpose some preliminary observations are in order.

In the fixed distribution setting, the dispersability condition is equivalent to the following requirement: As the cardinality of the partition Π approaches infinity, the sum over the elements F_i (forming the partition Π) of the average (with respect to Q) d_P-distance between the functions in F_i and some representative function f depending on F_i tends to zero. In mathematical terms, this requirement can be recast in the following statement equivalent to Definition 6. Denote by M the set of all maps $f\colon F \to F$ such that $f(g)$ is constant over F_i, $i = 1, \ldots, r$. Then the dispersability condition is equivalent to requiring that

$$\inf_{f \in M} E_Q[d_P(g, f(g))]$$

tends to zero when the size r of the partition $\Pi = \{F_i, i = 1, \ldots, r\}$ tends to infinity (compare with Definition 4). Extending this idea to a distribution-free setting requires some care. A straightforward, but rather naive, extension would consist in requiring that

$$\inf_{f \in M} \sup_{K \in \mathcal{K}^*} E_Q[d_{P(g)}(g, f(g))] \qquad (16)$$

tends to zero when the partition size r increases. However, a little thought reveals that sending the quantity in (16) to zero is in general an impossible task. Suppose for instance that we are considering concept learning. Then, the integrand $d_{P(g)}(g, f(g))$ can be always made equal to 1 by suitably selecting the probability $P(g)$, whenever $g \neq f(g)$.

The trouble with the above attempt to extend the definition of dispersability comes from the fact that one is asked to determine a partition able to reduce the dispersion, and yet, the metric $d_{P(g)}$ used to measure such a dispersion is unknown. Clearly this is an unfair game. To make the problem formulation more meaningful, the learner must be in a position to form some estimate of $P(g)$ before he is asked to determine the partition. This leads to the notion of *data dependent* partitions.

Consider a multisample $x = (x_1, \ldots, x_s)$. A partition Π of cardinality r based on the multisample x is simply a collection of partitions indexed by x:

$$\Pi = \{F_i(x), i = 1, \ldots, r\}.$$

Let M be the set of maps $f\colon X^s \times F \to F$ such that for all $x \in X^s$ and $g \in F$, $f(x, g)$ is constant over $F_i(x)$, $i = 1, \ldots, r$. The dispersion of the class F under partition Π is then defined as

$$\mathrm{disp}(\Pi) := \inf_{f \in M} \sup_{K \in \mathcal{K}^*} E_{\mathrm{Pr}_s}[d_{P(g)}(g, f(x, g))] \qquad (17)$$

where, in analogy with previous notation, Pr_s is defined as the product measure $Q \times [P(g)]^s$. The interpretation of (17) is as follows. Fix a map $f \in M$. Clearly $d_{P(g)}(g, f(x, g))$ is a random variable that depends on the multisample x and the target function g and it is therefore defined on $F \times X^s$. Such a random variable depends on the kernel K through $P(g)$. Next, the operator E_{Pr_s} performs integration over $F \times X^s$, thus returning the average distance of each g from the corresponding $f(x, g)$. The average here is with respect to the target function g and the random multisample x. So, all in all, $E_{\mathrm{Pr}_s}[d_{P(g)}(g, f(x, g))]$ is a deterministic number that measures the average dispersion of g from the corresponding $f(x, g)$; it depends on the map f and the kernel K. Finally, $\mathrm{disp}(\Pi)$ is defined as $\inf_f \sup_K E_{\mathrm{Pr}_s}[d_{P(g)}(g, f(x, g))]$ and, therefore, it quantifies how small such an average dispersion can be made in the worst case with respect to K by suitably selecting the map f.

Analogously to (16), Definition (17) is worst case owing to the presence of the quantifier $\sup_{K \in \mathcal{K}^*}$. However, differently from (16), in (17) the partition is allowed to depend on $x \in X^s$ and the dispersion is computed as an average over $F \times X^s$. Such a dependence gives one the possibility of forming some estimate of $P(g)$ before F is partitioned. Finally, the *minimal dispersion* $\mathrm{disp}(r, s)$ is defined as the infimum of $\mathrm{disp}(\Pi)$ when Π ranges over the set of all partitions of cardinality r based on the multisample $x \in X^s$.

Here, once again, a measurability issue arises. As a matter of fact, without any extra assumption on the map f, one cannot be sure that the function $d_{P(g)}(g, f(x, g))$ is measurable. Here and elsewhere in this section, we take the liberty of glossing over these measurability issues. This is a technical point certainly worthy of further investigation.

We are now in a position to define the notion of distribution-free dispersability.

Definition 8 (Distribution-Free Dispersability): The function class F is *distribution-free (d.f.) dispersable* if

$$\lim_{r, s \to \infty} \mathrm{disp}(r, s) = 0. \qquad \square$$

Note that $\mathrm{disp}(\cdot, \cdot)$ is a nonincreasing function of both arguments and, therefore, the order in which the limit $r, s \to \infty$ is taken in Definition 8 is immaterial. The fact that $\mathrm{disp}(\cdot, \cdot)$ is nonincreasing can be seen as follows. The function $d_{P(g)}(g, f(x, g))$ defined on $X^s \times F$ can be embedded in the larger invariant space $X^\infty \times F$. Then, $E_{\mathrm{Pr}_s}[d_{P(g)}(g, f(x, g))]$ becomes $E_{\mathrm{Pr}_\infty}[d_{P(g)}(g, f(x, g))]$, which exhibits no explicit dependence on s. Now, by increasing r and/or s, the set of maps M over which the infimum in (17) is taken becomes larger. It follows that $\mathrm{disp}(r, s)$ is a nonincreasing function of r and s.

B. Conditions for D.F. PAC Learning W.P.I.

In this section, various connections between the d.f. dispersability condition and the notion of d.f. PAC learning w.p.i.

are established. In the interest of clarity, the principal results are summarized beforehand.

1) Under the d.f. dispersability condition, a function class F is d.f. PAC learnable w.p.i.
2) The d.f. dispersability condition is satisfied if the Pollard-dimension of the function class F (or the Vapnik Chervonenkis-dimension in the case of a concept class C) is finite.
3) The d.f. dispersability condition is a *necessary and sufficient condition* for a concept class to be d.f. PAC learnable w.p.i.

Result 1 states that the d.f. dispersability condition is sufficient for the d.f. PAC learnability property. The second result brings out an interesting link between standard conditions for learning in a classical setting and the d.f. dispersability condition. Finally, the third result states the very interesting fact that the d.f. dispersability condition is a necessary and sufficient condition for the d.f. PAC learnability w.p.i. of a concept class C. In the light of this result, we can think of the d.f. dispersability condition as the *natural* condition for d.f. PAC learning a concept class w.p.i.

We begin by introducing an algorithm which generalizes Algorithm 1 of Section III so as to tailor it to a distribution-free framework.

Algorithm 3: Select two increasing integer-valued functions $r(t) \uparrow \infty$ and $s(t) \uparrow \infty$ such that $s(t) < t$ for all t.

At time t, do the following:

1) determine an optimal partition Π_t of cardinality $r(t)$ based on the multisample $x \in X^{s(t)}$, i.e., a partition Π_t such that $\mathrm{disp}(\Pi_t) = \mathrm{disp}(r(t), s(t))$;
2) determine a map f such that
$$\mathrm{disp}(\Pi_t) = \sup_{K \in \mathcal{K}^*} E_{Pr_s}[d_{P(g)}(g, f(x, g))];$$
3) compute the empirical error of each function $f_i(x)$, $i = 1, \ldots, r(t)$, associated with the map f, where $x = (x_1, \ldots, x_{s(t)})$ is the first $s(t)$-dimensional portion of the multisample $x = (x_1, \ldots, x_t)$
$$\hat{d}_{P(g),t}(g, f_i(x)) := \frac{1}{t - s(t)} \sum_{j=s(t)+1}^{t} |g(x_j) - f_i(x, x_j)|, \qquad i = 1, \ldots, r(t);$$
4) select
$$h_t := \operatorname*{argmin}_{f_i(x),\, i=1,\ldots,r(t)} \hat{d}_{P(g),t}(g, f_i(x)). \qquad \square$$

In Algorithm 3, the existence of an optimal partition Π_t and of a suitable map f is assumed. Should this be not the case, one can resort to suitable approximations.

The following theorem, which states that a function class is d.f. learnable w.p.i. provided that it is d.f. dispersable, is somehow expected.

Theorem 7: Suppose that the function class F is d.f. dispersable. If $s(t) = o(t)$ and $r(t) = \exp(o(t - s(t)))$, then Algorithm 3 d.f. PAC learns class F w.p.i.

Proof: The proof is an extension of those of Lemma 5 and Theorem 3.

First, fix arbitrarily a $g \in F$, a probability $P(g) \in \mathcal{P}^*$, a multisample $x \in X^{s(t)}$, and set
$$f^\circ := \operatorname*{argmin}_{f_i(x),\, i=1,\ldots,r(t)} d_{P(g)}(g, f_i(x)).$$
Similarly to Lemma 5, one can prove that
$$P(g)^{t-s(t)} \left\{ d_{P(g)}(g, h_t) - d_{P(g)}(g, f^\circ) \le \frac{\epsilon}{2} \right\} \ge 1 - (r(t) + 1) \exp[-(t - s(t))\epsilon^2/8], \qquad \forall \epsilon > 0. \tag{18}$$
Define
$$\Delta(g, P(g), x) := \left\{ x \in X^{t-s(t)} \colon d_{P(g)}(g, h_t) - d_{P(g)}(g, f^\circ) > \frac{\epsilon}{2} \right\}.$$
Then, for any kernel $K \in \mathcal{K}^*$, we have
$$\begin{aligned}
&\int_F \int_{X^{s(t)}} \int_{X^{t-s(t)} \setminus \Delta(g, P(g), x)} \left(d_{P(g)}(g, h_t) - \frac{\epsilon}{2} \right) \cdot P(g)^{t-s(t)}(dx) P(g)^{s(t)}(dx) Q(dg) \\
&\le \int_F \int_{X^{s(t)}} d_{P(g)}(g, f^\circ) P(g)^{s(t)}(dx) Q(dg) \\
&= E_{\Pr_{s(t)}}[d_{P(g)}(g, f^\circ)] \\
&\le \mathrm{disp}(\Pi_t) \\
&= \mathrm{disp}(r(t), s(t)).
\end{aligned} \tag{19}$$

Similarly to the derivation of (6), (19) used in conjunction with the estimate (18) gives
$$\Pr_t\{d_{P(g)}(g, h_t) > \epsilon\} \le (r(t) + 1) \exp(-(t - s(t))\epsilon^2/8) + \frac{2}{\epsilon} \mathrm{disp}(r(t), s(t)), \qquad \forall \epsilon > 0$$
for any kernel $K \in \mathcal{K}^*$.

From this, and also taking into account that $s(t) = o(t)$ and $r(t) = \exp(o(t - s(t)))$, the conclusion is immediately drawn that
$$\lim_{t \to \infty} \sup_{K \in \mathcal{K}^*} \Pr_t\{d_{P(g)}(g, h_t) > \epsilon\} = 0$$
that is, Algorithm 3 d.f. PAC learns w.p.i. the class F. $\square$

Our next result, besides being instrumental to the proof of Theorem 9 below, is of interest in its own right. It proves that the finiteness of the Vapnik Chervonenkis dimension implies the d.f. dispersability property.

Theorem 8: If VC-dimension $(C) < \infty$, then the concept class C is d.f. dispersable.

Proof: We start by introducing a "natural" partition of C of size $r = 2^s$ based on the multisample $x \in X^s$.

Given an integer $i \in [1, 2^s]$, denote by $b(i)$ the binary representation of $(i - 1)$. Let $C_i(x) \subset C$ be the collection of all sets A in C such that the jth element x_j of the multisample x belongs to A if and only if the jth digit of $b(i)$ is equal to 1. The so-called "natural" partition of C is then defined as
$$\Pi := \{C_i(x), i = 1, \ldots, 2^s\}.$$

Our goal consists in proving that

$$\lim_{s\to\infty} \mathrm{disp}(\Pi) = 0$$

which in turn implies that

$$\lim_{s\to\infty} \mathrm{disp}(2^s,\, s) = 0$$

which is the desired conclusion.

For any given $c \in C$ and $x \in X^s$ let $C_c(x)$ denote the element of the family $\{C_i(x),\, i = 1,\, \ldots,\, 2^s\}$ that contains c. We have

$$\begin{aligned}\mathrm{disp}(\Pi) &= \inf_{c' \in M} \sup_{K \in \mathcal{K}^*} E_{\Pr_s}[d_{P(c)}(c,\, c'(x,\, c))] \\ &\le \sup_{K \in \mathcal{K}^*} E_{\Pr_s}\left[\sup_{c' \in C_c(x)} d_{P(c)}(c,\, c')\right] \\ &\le \int_C \sup_{P \in \mathcal{P}^*} E_{P^s}\left[\sup_{c' \in C_c(x)} d_P(c,\, c')\right] Q\,(dc). \qquad (20)\end{aligned}$$

Next, it is shown that the integrand

$$\sup_{P \in \mathcal{P}^*} E_{P^s}\left[\sup_{c' \in C_c(x)} d_P(c, c')\right]$$

can be bounded above using the finiteness of the VC-dimension of C. Define

$$C\Delta C := \{c\Delta c' \colon c, c' \in C\}.$$

It is shown in [30, Th. 4.3, p. 88] that

$$\text{VC-dim}(C\Delta C) \le 10\text{VC-dim}(C) =: \eta, \text{ say.}$$

For any set $a \in C\Delta C$ and any multisample $x \in X^s$, define its *empirical probability* as

$$\hat{P}(a;\, x) := \frac{1}{s}\sum_{i=1}^{s} I_a(x_i).$$

Then it is known (see [30, Th. 7.2, p. 198]) that

$$\begin{aligned}&P^s\left\{x \in X^s \colon \sup_{a \in C\Delta C} |\hat{P}(a;\, x) - P(a)| > \epsilon\right\} \\ &\qquad \le 4\left(\frac{2es}{\eta}\right)^{\eta} \exp(-s\epsilon^2/8) =: \nu(s,\, \epsilon), \text{ say.}\end{aligned}$$

In particular, recall that $C_c(x)$ denotes the particular component of the partition of C that contains c. Hence

$$\hat{P}(c\Delta c';\, x) = 0, \qquad \forall\, c' \in C_c(x).$$

Therefore

$$P^s\left\{x \in X^s \colon \sup_{c \in C}\, \sup_{c' \in C_c(x)} d_P(c,\, c') > \epsilon\right\} \le \nu(s, \epsilon).$$

This implies that the integrand in (20) satisfies

$$\sup_{P \in \mathcal{P}^*} E_{P^s}\left[\sup_{c' \in C_c(x)} d_P(c, c')\right] \le [1 - \nu(s,\, \epsilon)]\epsilon + \nu(s,\, \epsilon).$$

Since $\nu(s,\, \epsilon)$ approaches zero as s approaches infinity for each fixed ϵ, it can be seen that this quantity can be made arbitrarily small by choosing s large enough. Hence $\mathrm{disp}(\Pi) \to 0$, as $s \to \infty$. □

Using entirely analogous reasoning, Theorem 8 can be extended to function classes with finite Pollard-dimension. In particular, any function class with finite Pollard-dimension is d.f. dispersable. Moreover, the so-called "natural partition" defined above can be readily extended to a family of functions, and it disperses the function family. As a consequence, it follows that the procedure described in Algorithm 3 is PAC w.p.i. even for a function family.

Together, Theorems 7 and 8 establish that finite VC- (or Pollard-) dimension implies d.f. dispersability, which in turn implies d.f. learnability w.p.i. This conclusion is not surprising, since the finiteness of the VC- or Pollard-dimension in fact implies the much stronger property of d.f. learnability (without prior information).

The fact that $P(g,\, A)$ must be $\mathcal{F}$-measurable for any $A \in \mathcal{X}$ imposes a constraint on permissible kernels. In the next theorem, we show that, in the case of concept classes, if the set of kernels $\mathcal{K}^*$ is rich enough that any family of probabilities $\{P(g),\;\; g \in F\}$ is in fact a permissible kernel, then d.f. dispersability is a *necessary and sufficient* condition for d.f. learnability w.p.i. Thus, d.f. dispersability characterizes d.f. learnability w.p.i., in the same way that finite VC-dimension characterizes d.f. learnability (without prior information).

Theorem 9: If $\mathcal{K}^*$ is the set of all families of probabilities $P(g)$ indexed by $g \in F$, then a function class C is d.f. PAC learnable w.p.i. if and only if it is d.f. dispersable.

Proof: The "if" part is a straightforward consequence of Theorems 7 and 8. Therefore, we concentrate here on the "only if" part.

Consider the product space $C \times X^\infty$ and for a given kernel K, denote by Pr_K the corresponding probability measure in $C \times X^\infty$. In this framework, the fact that the algorithm $\{a_t\}$ d.f. PAC learns w.p.i. the concept class C translates into the condition

$$\lim_{t\to\infty} \sup_{K \in \mathcal{K}^*} \mathrm{Pr}_K\left\{d_{P(c)}(c,\, h_t) > \epsilon\right\} = 0, \qquad \forall\, \epsilon > 0. \quad (21)$$

Our intermediate goal consists in showing that (21) implies that for all $\rho > 0$, there exists a set $C(\rho) \subset C$ such that

1) $Q(C(\rho)) \ge 1 - \rho$;
2) $\lim_{n\to\infty} \sup_{c \in C(\rho)} \sup_{P \in \mathcal{P}^*} P^\infty\{d_P(c,\, h_{t_n}) > \epsilon\} = 0, \forall\, \epsilon > 0$;

where $\{t_n\}$ is a suitable sequence of time instants.

First, fix a sequence of real numbers $\epsilon_n \downarrow 0$. From (21) it is easy to see that a sequence $\{t_n\}$ of time instants can be determined such that

$$\sum_{n=1}^{\infty} \sup_{K \in \mathcal{K}^*} Pr_K\left\{d_{P(c)}(c, h_{t_n}) > \epsilon_n\right\} < \infty.$$

This implies that

$$\sum_{n=1}^{\infty} \int_C \sup_{P \in \mathcal{P}^*} P^\infty\{d_P(c,\, h_{t_n}) > \epsilon_n\} Q\,(dc) < \infty$$

from which it follows that:

$$\lim_{n\to\infty} \sup_{P\in\mathcal{P}^*} P^\infty\{d_P(c, h_{t_n}) > \epsilon_n\} = 0, \quad Q\text{—almost surely}$$

that is, $\sup_{P\in\mathcal{P}^*} P^\infty\{d_P(c, h_{t_n}) > \epsilon_n\}$ converges to zero almost surely as a function of c. Now it is well known that if a random variable converges almost surely to zero, then it converges uniformly to zero on a subset whose probability is arbitrarily close to 1. Thus for all $\rho > 0$, there exists a set $C(\rho) \subset C$ such that

i) $Q(C(\rho)) \geq 1 - \rho$;

ii) $\lim_{n\to\infty} \sup_{c\in C(\rho)} \sup_{P\in\mathcal{P}^*} P^\infty\{d_P(c, h_{t_n}) > \epsilon_n\} = 0$.

Condition i) is the same as Condition 1), whereas Condition ii) obviously implies Condition 2) since $\epsilon_n \downarrow 0$. Thus our intermediate goal is established.

Note now that Condition 2 corresponds to the requirement that the algorithm $\{a_{t_n}\}$ d.f. PAC learns the concept class $C(\rho)$. On the other hand, it is well known, [13], that this requirement is equivalent to the finiteness of the VC-dimension of the class $C(\rho)$. Thus it has been shown that for all $\rho > 0$ there exists a set $C(\rho) \subset C$ such that

1) $Q(C(\rho)) \geq 1 - \rho$ and

3) $\text{VC-dim}(\rho)) < \infty$.

The dispersability property of C can now be easily proven from Statements 1 and 3, using Theorem 8. Consider a sequence of partitions $\Pi_{r,s}^{C(\rho)}$ of $C(\rho)$ based on the multisample $x \in X^s$ of cardinality r such that

$$\lim_{r,s\to\infty} \operatorname{disp}\left(\Pi_{r,s}^{C(\rho)}\right) = 0. \tag{22}$$

Such a sequence exists in view of Theorem 8 and Statement 3. Then partition C as follows:

$$\Pi_{r+1,s} := \Pi_{r,s}^{C(\rho)} \cup \overline{C}(\rho).$$

Then, we have

$$\begin{aligned}\operatorname{disp}(\Pi_{r+1,s}) &\leq \operatorname{disp}\left(\Pi_{r,s}^{C(\rho)}\right) + \rho \,(\text{using Statement 1})\\ &\to \rho \quad \text{as} \quad r, s \to \infty. \text{ [using (22)]}\end{aligned}$$

Since ρ is arbitrary, this implies that $\operatorname{disp}(\Pi_{r+1,s}) \to 0$ and the theorem is proved. □

The proof of Theorem 9 suggests that, for *concept* classes, a variant of Algorithm 3 is possible.

Algorithm 4: Select an increasing integer function $s(t) \uparrow \infty$. At time t, do the following:

1) determine the natural partition $\Pi_t := \{C_i(x)\}_{i=1}^{2^{s(t)}}$, where $x = (x_1, \ldots, x_{s(t)})$ is the first $s(t)$-dimensional portion of the multisample $x = (x_1, \ldots, x_t)$;

2) for $i = 1, \ldots, 2^{s(t)}$, extract at random a concept $c_i(x)$ out of $C_i(x)$ according to probability Q restricted to $C_i(x)$;

3) compute the empirical error of each concept $c_i(x)$:

$$\hat{d}_{P(c),t}(c, c_i(x)) := \frac{1}{t - s(t)} \sum_{j=s(t)+1}^{t} |c(x_j) - c_i(x, x_j)|, \quad i = 1, \ldots, 2^{s(t)};$$

4) select

$$h_t := \operatorname*{argmin}_{c_i(x), i=1,\ldots,2^{s(t)}} \hat{d}_{P(c),t}(c, c_i(x)).$$ □

It is important to note that the *random* extraction of concepts $\{c_i(x)\}$ at the second step of Procedure 2 has a fundamental beneficial effect which is missing if random extraction is replaced by a deterministic selection. This can be intuitively explained as follows. The concept class C may contain a subset of overly complex concepts [and, in fact, *VC-dimension* (C) may well be infinite]. However, if this is the case, such a "pathological" subset will have a negligible probability Q (see Statement 1 in the proof of Theorem 9). Therefore, a random extraction of concepts $\{c_i(x)\}$ will fall in the pathological subset with negligible probability. This advantage is obviously missing if deterministic selection is used, since the probability of falling into the pathological subset is no longer governed by Q.

We end this section with a final theorem concerning Algorithm 4. Its proof is omitted, but can be easily worked out based on the observations presented so far. Also, in the statement of the theorem we have glossed over the tedious definition of probability $\Pr_t$.

Theorem 10: Choose $s(t)$ such that $2^{s(t)} = o(t)$. If the concept class C is d.f. PAC-learnable w.p.i., then h_t computed through Algorithm 4 is such that

$$\lim_{t\to\infty} \sup_{K\in\mathcal{K}^*} \Pr_t\{d_{P(c)}(c, h_t) > \epsilon\} = 0, \qquad \forall \epsilon > 0$$

where $\Pr_t$ is a probability which accounts for all random elements in the problem, i.e., $c \in C$, $x \in X^t$ and the random selection of concepts $c_i(x)$. □

VI. Concluding Remarks

In this paper, we have introduced a new notion of learning, called learning with prior information. This new notion of learnability is significantly weaker than the widely studied notion of PAC (probably approximately correct) learnability. Necessary and sufficient conditions have been derived for a concept class to be learnable with prior information, both in fixed-distribution learning and distribution-free learning. A new concept called "dispersability" has been introduced, and it has been shown that dispersability (defined appropriately for the situation) is both necessary and sufficient for learnability with prior information. Thus, the results presented here are quite definitive.

It has also been shown that *any* collection of measurable functions mapping a separable metric space into a compact interval is learnable with prior information. This result, while conclusive and elegant, also suggests that perhaps learnability with prior information is too mild a form of learnability. Thus there still remains the challenging problem of defining other, still newer notions of learnability that are intermediate between PAC learnability and learnability with prior information. This is a topic for further research.

Acknowledgment

The first author would like to thank M. Vidyasagar for his warm hospitality during his visit in Bangalore.

References

[1] A. R. Barron, “Universal approximation bounds for superpositions of a sigmoidal function,” *IEEE Trans. Inform. Theory*, vol. 39, pp. 930–945, Mar. 1993.

[2] P. L. Bartlett, P. M. Long, and R. C. Williamson, “Fat-shattering and the learnability of real-valued functions,” *J. Comp. Sys. Sci.*, vol. 52, no. 3, pp. 534–552, 1996.

[3] G. M. Benedek and A. Itai, “Learnability by fixed distribution,” in *Proc. 1st Workshop Computational Learning Theory*, 1988, pp. 80–90.

[4] ——, “Nonuniform learnability,” *J. Comp. Syst. Sci.*, vol. 48, pp. 311–323, 1994.

[5] P. Billingsley, “Probability and measure,” in *Wiley Series in Probability and Mathematical Statistics*, 3rd ed. New York: Wiley, 1995.

[6] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. Warmuth, “Learnability and the Vapnik–Chervonenkis dimension,” *J. ACM*, vol. 36, pp. 929–965, 1989.

[7] L. Breiman, “Hinging hyperplanes for regression, classification and function approximation,” *IEEE Trans. Inform. Theory*, vol. 39, pp. 999–1013, Mar. 1993.

[8] K. L. Buescher and P. R. Kumar, “Learning by canonical smooth estimation—Part I: Simultaneous estimation,” *IEEE Trans. Automat. Contr.*, vol. 42, pp. 545–556, Apr. 1996.

[9] ——, “Learning by canonical smooth estimation—Part II: Learning and choice of model complexity,” *IEEE Trans. Automat. Contr.*, vol. 42, pp. 557–569, Apr. 1996.

[10] V. Cherkassky and F. Mullier, *Learning From Data*. New York: Wiley, 1998.

[11] L. Devroye, L. Gyorfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*. New York: Springer-Verlag, 1996.

[12] R. M. Dudley, *A Course in Empirical Processes*, ser. Lecture Notes in Mathematics, 1984, vol. 1097.

[13] A. Ehrenfeucht, D. Haussler, M. Kearns, and L. Valiant, “A general lower bound on the number of examples needed for learning,” *Inform. Comp.*, vol. 82, pp. 247–261, 1989.

[14] A. Fraenkel and Y. Bar-Hillel, *Foundations of Set Theory*. Amsterdam, The Netherlands: Noth Holland, 1958.

[15] U. Grenander, “Abstract inference,” in *Wiley Series in Probability and Mathematical Statistics*. New York: Wiley, 1981.

[16] D. Haussler, “Decision theoretic generalizations of the PAC model for neural net and other learning applications,” *Inform. Comp.*, vol. 100, pp. 78–150, 1992.

[17] D. Haussler, M. Kearns, and R. Schapire, “Bounds on the sample complexity of Bayesian learning using information theory and the VC dimension,” in *Proc. Conf. Comp. Learning Theory*, 1991, pp. 61–74.

[18] D. Haussler, M. Kearns, M. Opper, and R. Schapire, “Estimating average-case learning curves using Bayesian, statistical physics and VC dimension methods,” *Adv. Neural Inform. Processing*, pp. 855–862, 1992.

[19] A. N. Kolmogorov and V. M. Tikhomirov, “ϵ-entropy and ϵ-capacity of sets in functional spaces,” *Americ. Math. Soc. Transl.*, vol. 17, pp. 277–364, 1961.

[20] D. A. McAllester, *Some PAC-Bayesian Theorems*, J. Baxter, Ed. Boston, MA: Kluwer, 1999, pp. 355–363.

[21] ——, “PAC-Bayesian theorems,” in *Proc. Conf. Comput. Learning Theory*, 1999.

[22] N. Norgaard, O. Ravn, N. K. Poulsen, and L. K. Hansen, *Neural Networks for Modeling and Control of Dynamic Systems*. New York: Springer-Verlag, 2000.

[23] D. Pollard, *Convergence of Stochastic Processes*. New York: Springer-Verlag, 1984.

[24] ——, “Empirical processes: Theory and applications,” in *Regional Conference Series in Probability and Statistics*, vol. 2, 1990.

[25] J. Shawe-Taylor and R. Williamson, “A PAC analysis of a Bayesian estimator,” in *Proc. Conf. Comp.Learning Theory*, 1997.

[26] L. G. Valiant, “A theory of the learnable,” *Commun. ACM*, vol. 27, pp. 1134–1142, 1984.

[27] V. N. Vapnik and A. Y. Chervonenkis, “Uniform convergence of the frequencies of occurrence of events to their probabilities,” *Soviet Math. Doklady*, vol. 9, pp. 915–918, 1968.

[28] ——, “On the uniform convergence of relative frequencies to their probabilities,” *Theory Probab. Appl.*, vol. 16, pp. 264–280, 1971.

[29] ——, “Necessary and sufficient conditions for the uniform convergence of means to their expectations,” *Theory Probab. Appl.*, vol. 26, pp. 532–553, 1981.

[30] M. Vidyasagar, *A Theory of Learning and Generalization*. New York: Springer-Verlag, 1997.

Marco C. Campi was born in Tradate, Italy, in 1963. He received the Doctor degree in electronic engineering from the Politechnico di Milano, Milan, Italy, in 1988.

From 1988 to 1989, he was a Reserach Assistant in the Department of Electrical Engineering, the Politechnico di Milano. From 1989 to 1992, he worked as a researcher at the Centro di Teoria dei Sistemi of the National Research Council in Milan, Italy. Since 1992, he has been with the University of Brescia, Italy, where he is currently Professor of Automatic Control. He has held visiting positions at many universities and institutions, including the Australian National University, Canberra, Australia, the University of Illinois at Urbana-Champaign, USA, the Centre for Artificial Intelligence and Robotics, Bangalore, India, and the University of Melbourne, Australia. He is an Associate Editor of *Automatica* and of the *European Journal of Control*. His current research interests include: learning theory, adaptive and iterative control, system identification, and stochastic systems.

Dr. Campi is Vice-Chair of the Technical Committee IFAC on Stochastic Systems (SS) and a member of the Technical Committee IFAC on Modeling, Identification, and Signal Processing (MISP). He is a Distinguished Lecturer under the IEEE CSS Program.

M. Vidyasagar (S’69–M’69–SM’78–F’83) was born in Guntur, Andhra Pradesh, India in 1947. He received the B.S., M.S., and Ph.D. degrees, all in electrical engineering, from the University of Wisconsin, in 1965, 1967, and 1969, respectively.

He has taught at Marquette University, Milwaukee, WI, (1969-70), Concordia University, Canada (1970-80), and the University of Waterloo, Canada (1980-89). In 1989, he returned to India as the Director of the Centre for Artificial Intelligence and Robotics, (under the Defence Research and Development Organisation), Bangalore. In 2000, he took up his current assignment as Executive Vice President (Advanced Technology) in Tata Consultancy Services, which is India’s largest IT firm. Currently, he is based in the city of Hyderabad. In his current position, his responsibilities are to create an Advanced Technology Centre (ATC) within TCS, to develop futuristic technologies of relevance to the IT industry. At the present time, the scope of activities of the ATC includes PKI (Public Key Infrastructure), security in e- and m-commerce, and advanced cryptography including elliptic curve cryptography, and neural networks. Most recently, the ATC has also undertaken a major initiative in the emerging area of bioinformatics.\\ In the past, he has held visiting positions at several universities including the Massachusetts Institute of Technology, Cambridge, MA, the University of California, Berkeley, the University of California, Los Angeles, C.N.R.S. Toulouse, France, Indian Institute of Science, University of Minnesota, and the Tokyo Institute of Technology, Tokyo, Japan. He is the author or co-uthor of seven books and more than one hundred and twenty papers in archival journals. His current research interests are control theory, machine learning and its applications to bioinformatics, and elliptic-curve cryptography.

Dr. Vidyasagar has received several honors in recognition of his research activities, including the Distinguished Service Citation from his Alma Mater (The University of Wisconsin), and the IEEE Hendrik W. Bode Lecture Prize for the year 2000. In addition, he is a Fellow of the Indian Academy of Sciences, the Indian National Science Academy, the Indian National Academy of Engineering, and the Third World Academy of Sciences.