Journal of Machine Learning Research 10 (2009) 1095-1129 Submitted 3/08; Revised 11/08; Published 5/09

Universal Kernel-Based Learning with Applications to Regular Languages

Leonid (Aryeh) Kontorovich ARYEH.KONTOROVICH@WEIZMANN.AC.IL
Department of Mathematics
Weizmann Institute of Science
Rehovot, Israel 76100

Boaz Nadler BOAZ.NADLER@WEIZMANN.AC.IL
Department of Computer Science and Applied Mathematics
Weizmann Institute of Science
Rehovot, Israel 76100

Editors: Paolo Frasconi, Kristian Kersting, Hannu Toivonen and Koji Tsuda

Abstract

We propose a novel framework for supervised learning of discrete concepts. Since the 1970's, the standard computational primitive has been to find the most consistent hypothesis in a given complexity class. In contrast, in this paper we propose a new basic operation: for each pair of input instances, count how many concepts of bounded complexity contain both of them.

Our approach maps instances to a Hilbert space, whose metric is induced by a universal kernel coinciding with our computational primitive, and identifies concepts with half-spaces. We prove that all concepts are linearly separable under this mapping. Hence, given a labeled sample and an oracle for evaluating the universal kernel, we can efficiently compute a linear classifier (via SVM, for example) and use margin bounds to control its generalization error. Even though exact evaluation of the universal kernel may be infeasible, in various natural situations it is efficiently approximable.

Though our approach is general, our main application is to regular languages. Our approach presents a substantial departure from current learning paradigms and in particular yields a novel method for learning this fundamental concept class. Unlike existing techniques, we make no structural assumptions on the corresponding unknown automata, the string distribution or the completeness of the training set. Instead, given a labeled sample our algorithm outputs a classifier with guaranteed distribution-free generalization bounds; to our knowledge, the proposed framework is the only one capable of achieving the latter. Along the way, we touch upon several fundamental questions in complexity, automata, and machine learning.

Keywords: grammar induction, regular language, finite state automaton, maximum margin hyperplane, kernel approximation

1. Introduction

We begin by describing the basic problem setting and outlining our approach.

1.1 Background

Perhaps the most fundamental problem in learning from labeled data is to construct a classifier with guaranteed small generalization error. Typically, given labeled data there exists a nested sequence of candidate concept classes with increasing complexity. Of course, a classifier from a rich concept class can easily achieve a low or even zero training error, but is likely to overfit the data and have a poor generalization performance. Hence, a major challenge is to choose the hypothesis class with appropriate complexity. This issue also arises in classical statistics, where it is known as the *model selection* problem (Rissanen, 1989). In Vapnik (1982), the structural risk minimization (SRM) principle was proposed to solve this problem, by quantifying the bias-variance tradeoff between fit to the data and model complexity, as captured by the VC-dimension.

The basic computational primitive in the SRM framework is finding the most consistent hypothesis in a given concept class. Although SRM resolves the information theoretic problem of model selection, it offers no method for finding such a hypothesis. Moreover, for discrete concept classes, finding this hypothesis may be infeasible, as this computational primitive typically requires solving a hard combinatorial optimization problem.

An important instance of this scenario is the problem of learning a regular language from labeled examples. Supervised language learning from examples is one of the most fundamental problems in learning theory, and as such has fascinated philosophers, linguists, computer scientists and mathematicians as a quintessential problem of induction. Insofar as the PAC model (Valiant, 1984) is a natural formalization of learning and the regular languages are the simplest nontrivial class of formal languages, the vast amount of literature devoted to learning regular languages is well justified (see Rivest and Schapire 1987 and de la Higuera 2005 and the references therein).

Applying the standard SRM principle to learning a regular language requires finding the smallest automaton consistent with the given labeled sample. However, it was realized early on that this task is computationally very difficult. Finding the smallest automaton consistent with a set of accepted and rejected strings was shown to be NP-complete by Angluin (1978) and Gold (1978); this was further strengthened in the hardness of approximation result of Pitt and Warmuth (1993), where it was proven that even finding a DFA with a number of states polynomial in the number of states of the minimum solution is NP-complete.

To make matters worse, it turns out that under cryptographic assumptions the class of regular languages is inherently hard to learn, regardless of representation. Thus, assuming the average-case difficulty of the Discrete Cube Root problem, Theorem 7.6 of Kearns and Vazirani (1997) shows how to construct small automata over $\{0,1\}$ and distributions over $\{0,1\}^n$ for which it is intractable to discover a hypothesis with error less than $1/2-1/p(n)$, for any fixed polynomial p. A similar construction shows that this task is also at least as hard as factoring.

Nevertheless, a number of positive learnability results were obtained under various restricted settings. Trakhtenbrot and Barzdin (1973) showed that the smallest finite automaton consistent with the input data can be learned exactly from a complete sample of all strings up to a given length. The worst case complexity of their algorithm is exponential in automaton size, but a better average-case complexity can be obtained assuming that the topology and the labeling are selected randomly or even that the topology is selected adversarially (Freund et al., 1993). In a different model of learnability—“identification in the limit” (Gold, 1967)—positive results were obtained for the k-reversible languages (Angluin, 1982) and subsequential transducers (Oncina et al., 1993). Some restricted classes of probabilistic automata such as acyclic probabilistic automata were also shown to be efficiently learnable by Ron et al. (1998). In a modified model of PAC, and with additional language complexity constraints, Clark and Thollard (2004) showed a class of probabilistic finite state automata to be learnable; see also literature review therein.

In light of the strong negative results mentioned above, the prevailing paradigm in formal language learning has been to make structural regularity assumptions about the family of languages and/or the sampling distribution in question and to employ a state-merging heuristic. Indeed, over the years a number of clever and sophisticated combinatorial approaches have been proposed for learning DFAs. Typically, an initial automaton or prefix tree consistent with the sample is first created. Then, starting with the trivial partition with one state per equivalence class, classes are merged while preserving an invariant congruence property. The automaton learned is obtained by merging states according to the resulting classes. Thus, the choice of the congruence determines the algorithm and generalization bounds are obtained from the structural regularity assumptions. This rough summary broadly characterizes the techniques of Angluin (1982); Oncina and Garcia (1992); Ron et al. (1998); Clark and Thollard (2004) and until recently, this appears to have been the only general-purpose technique available for learning finite automata.

Finally, despite the fact that the problems of inducing a regular language, and specifically finding the smallest automaton consistent with a given sample were proven to be theoretically computationally hard, in recent years various heuristic search methods have been developed with reported considerable success in solving these problems for moderately sized DFAs, see for example Lang (1992), and Oliveira and Silva (2001) and Bugalho and Oliveira (2005). Of course, these heuristic methods find some automaton consistent with the given sample, but provide no generalization bounds on its expected performance on new samples.

1.2 Learning Languages Via Hyperplanes

In this paper we propose a novel framework for learning a regular language from a finite sample of positive and negative labeled strings. Rather than attempting to find a *single* small consistent automaton, we embed the strings in a Hilbert space and compute a maximum margin hyperplane, which becomes our classifier for new strings. Hence, we effectively convert a difficult combinatorial problem into a (high dimensional) geometric one, amenable to efficient techniques using linear algebra and convex optimization. In particular, our resulting classifier is a linear combination of potentially smaller automata, each of which is typically not consistent with the training data.

Since the advent of Support Vector Machines (SVMs), maximum margin classifiers have flourished in the machine learning community (Schölkopf and Smola, 2002). Within the context of grammatical inference, the first work to apply this methodology was Kontorovich et al. (2006), where a kernel was used to learn a specific family of regular languages (the piecewise-testable ones). The authors embed the set of all strings onto a high-dimensional space and achieve language induction by constructing a maximum-margin hyperplane. This hinges on every language in a family of interest being *linearly separable* under the embedding, and on the efficient computability of the kernel. This line of research is continued in Cortes et al. (2007), where linear separability properties of rational kernels are investigated.

In this paper, we build upon the approach suggested in Kontorovich (2007), where a *universal* kernel is given that renders *all* regular languages linearly separable. We prove that any linearly separable language necessarily has a positive margin, so standard margin-based generalization guarantees apply. A by-product of this result is a novel characterization of the regular languages as precisely those that are linearly separable under our kernel. A drawback of this kernel is that a brute-force computation is infeasible, while an efficient one is currently unknown (and conjectured not to exist unless P= #P). However, we propose a simple randomized scheme for efficiently obtaining an ε-approximation to our universal kernel. We show that the approximate kernel preserves the distances with low distortion, and therefore may be used as a surrogate for the original one. Random sampling of embedding features to evaluate otherwise intractable kernels seems to first have been employed in Kontorovich (2007); Rahimi and Recht (2007) used a similar idea in a somewhat different setting. Also related are the approaches of Garg et al. (2002) and Balcan et al. (2006), where classifiers and their generalization performance are computed based on low-dimensional random projections. Note the marked difference between our method and theirs: we take a random subset of the embedding features as opposed to projecting the embedding space onto a random subspace; in fact, the setting we consider does not seem to allow the use of random projections. Also, note that in general, projecting the sample onto a low-dimensional subspace does not guarantee linear separability; a somewhat delicate margin analysis is needed, as described in Section 5.

From a practical standpoint, our resulting classifier is the sign of a weighted linear combination of possibly many random DFAs. The fact that a complicated DFA with many states may be written as a weighted sum of much simpler DFAs is illustrated below and provides additional motivation for our approach. Our methodology is intimately connected to boosting (see, for example, Bartlett et al. 1998), since our resulting classifier can also be viewed as a weighted sum of weak classifiers—each being one of the random DFAs. Furthermore, our approach has a natural interpretation in the structural risk minimization (SRM) framework, where we trade the computational problem of finding a small consistent hypothesis for the problem of counting the number of small concepts accepting a given instance. The advantage of the latter is that it admits an efficient approximation in many cases of interest, including the case of DFAs.

In summary, our proposed algorithm is efficient, simple to implement, comes with strong theoretical guarantees, and readily generalizes to many other learning settings, such as context-free languages. We also present preliminary empirical results as a proof of concept. To our knowledge, the framework we propose is the only one capable of inducing unrestricted regular languages from labeled samples, without imposing additional structural assumptions—modulo standard cryptographic limitations. Some of the results in this paper first appeared in the conference version (Kontorovich, 2007).

1.3 Outline of Paper

This paper is organized as follows. In Section 2 we set down the notation and definitions used throughout the paper. The notion of linearly separable concept classes is defined in Section 3, and a general theorem regarding their existence via a canonical embedding is proved in Section 4. In Section 5, the universal kernel oracle is used to obtain an efficient generic learning algorithm, and in Section 6 an efficient approximation scheme is given for computing the universal kernel, with a corresponding learning algorithm. In Section 7, these results are applied to the case of the regular languages, with some experimental results presented in Section 8. Some inherent limitations of such kernel-based techniques are discussed in Section 9, and concluding remarks are made in Section 10.

2. Preliminaries

We refer the reader to Kearns and Vazirani (1997) for standard learning-theoretic definitions such as *instance space* and *concept class*. We likewise use standard set-theoretic terminology, with $|\cdot|$ denoting set cardinalities and $\mathbb{1}_{\{\cdot\}}$ representing the 0-1 truth value of the predicate inside the brackets. We blur the distinctions between subsets $c \subset \mathcal{X}$ and binary functions $c : \mathcal{X} \to \{0, 1\}$; when we wish to make the distinction explicit, we will use a $\{\pm 1\}$-valued characteristic function:

$$\chi_c(x) := 2\mathbb{1}_{\{x \in c\}} - 1.$$

Recall that if A and B are two sets, B^A represents the collection of all functions from A to B. For countable sets A, we will consider the vector space $\mathbb{R}^A$ and often index $x \in \mathbb{R}^A$ by elements of A as opposed to by the naturals. For example, for $x, y \in \mathbb{R}^A$, we define their *inner product* as

$$\langle x, y \rangle \quad = \quad \sum_{a \in A} [x]_a [y]_a,$$

whenever the latter sum is well-defined. Square brackets may be omitted from the indexing notation for readability.

For any $x \in \mathbb{R}^{\mathbb{N}}$, we define its *support* to be the set of its nonzero coordinates:

$$\mathrm{supp}(x) \quad = \quad \{i \in \mathbb{N} : x_i \neq 0\}.$$

We define the quasinorm $\|\cdot\|_0$ to be the number of the nonzero coordinates:

$$\|x\|_0 \quad = \quad |\mathrm{supp}(x)|,$$

and the standard ℓ_2 norm by $\|x\|_2^2 = \langle x, x \rangle$.

If V_1 and V_2 are two inner product spaces over $\mathbb{R}$, we define their *direct product* $V_1 \otimes V_2$ to be the vector space

$$V_1 \otimes V_2 \quad = \quad \{(x, y) : x \in V_1, y \in V_2\}$$

which naturally inherits vector addition and scalar multiplication from V_1 and V_2, and whose inner product is defined by

$$\left\langle (x, y), (x', y') \right\rangle \quad = \quad \left\langle x, x' \right\rangle + \left\langle y, y' \right\rangle.$$

We will also write $(x, y) \in V_1 \otimes V_2$ as $x \otimes y$. Note that $\|x \otimes y\|_0 = \|x\|_0 + \|y\|_0$.

3. Linearly Separable Concept Classes

Our main results are actually quite general and most of them make no use of the properties of regular languages and automata. The only property we use is the *countability* of the instance space $\mathcal{X}$ and of the concept class $\mathcal{C}$. Henceforth, $\mathcal{X}$ and $\mathcal{C}$ are always assumed to be (at most) countable, unless noted otherwise. Let $\mathcal{C}$ be a concept class defined over an instance space $\mathcal{X}$; thus, $\mathcal{C} \subseteq 2^{\mathcal{X}}$. Any $c \in \mathcal{C}$ is thus a classifier or a function that induces a $\{0, 1\}$ labeling on the instance space $\mathcal{X}$. Note that a concept $c \subset \mathcal{X}$ may contain infinitely many instances, and an instance $x \in \mathcal{X}$ may belong to an infinite number of concepts.

We will say that a concept $c \in \mathcal{C}$ is *finitely linearly separable* if there exists a mapping $\phi : \mathcal{X} \to \{0, 1\}^{\mathbb{N}}$ and a weight vector $w \in \mathbb{R}^{\mathbb{N}}$, both with *finite support*, that is, $\|w\|_0 < \infty$ and $\|\phi(x)\|_0 < \infty$ for all $x \in \mathcal{X}$, such that

$$c = \{x \in \mathcal{X} : \langle w, \phi(x) \rangle > 0\}.$$

The concept class $\mathcal{C}$ is said to be *finitely linearly separable* if all $c \in \mathcal{C}$ are finitely linearly separable under the *same* mapping ϕ.

Note that the condition $\|\phi(\cdot)\|_0 < \infty$ is important; otherwise, we could define the *embedding by concept* $\phi : \mathcal{X} \to \{0,1\}^{\mathcal{C}}$

$$[\phi(x)]_c \quad = \quad \mathbb{1}_{\{x \in c\}}, \qquad c \in \mathcal{C}$$

and for any target $\hat{c} \in \mathcal{C}$,

$$[w]_c \quad = \quad \mathbb{1}_{\{c = \hat{c}\}}.$$

This construction trivially ensures that

$$\langle w, \phi(x) \rangle \quad = \quad \mathbb{1}_{\{x \in \hat{c}\}}, \qquad x \in \mathcal{X},$$

but may not satisfy $\|\phi(\cdot)\|_0 < \infty$ since certain $x \in \mathcal{X}$ may belong to an infinite number of concepts.

Similarly, we disallow $\|w\|_0 = \infty$ due to the algorithmic impossibility of storing infinitely many numbers and also because it leads to the trivial construction, via *embedding by instance* $\phi : \mathcal{X} \to \{0,1\}^{\mathcal{X}}$

$$[\phi(x)]_u \quad = \quad \mathbb{1}_{\{x = u\}}, \qquad u \in \mathcal{X},$$

whereby for any target $\hat{c} \in \mathcal{C}$, the corresponding vector

$$[w]_u \quad = \quad \mathbb{1}_{\{u \in \hat{c}\}}$$

ensures $\langle w, \phi(x) \rangle = \mathbb{1}_{\{x \in \hat{c}\}}$ without doing anything interesting or useful.

An additional important reason to insist on finite linear separability is that it ensures that for each $c \in \mathcal{C}$ there is a "separability dimension" $D(c) < \infty$ such that c is linearly separable in $\{0,1\}^{D(c)}$ under ϕ (in particular, if w defines a separating hyperplane for c then $D(c) \le \|w\|_0$). Now any linearly separable partition of $\{0,1\}^D$ must necessarily be separable with a strictly positive margin; this is true on any discrete finite space (but false in $\{0,1\}^\infty$). The latter in turn provides generalization guarantees, as spelled out in Section 5. Finally, any embedding ϕ induces a kernel K via

$$K(x,y) \quad = \quad \langle \phi(x), \phi(y) \rangle \tag{1}$$

for $x,y \in \mathcal{X}$. The finite support of ϕ automatically guarantees that K is everywhere well-defined; otherwise, additional assumptions are needed.

In light of the discussion above, from now on the only kind of linear separability we shall consider is in the strong sense defined above, where both the embedding $\phi(\cdot)$ and the hyperplane vector w have finite support. The modifiers "countable" (for instance spaces and concept classes), "finitely" (for linear separability) and "finite support" (for embeddings and hyperplanes) will be implicitly assumed throughout. Since an embedding $\phi : \mathcal{X} \to \{0,1\}^{\mathbb{N}}$ induces a kernel K as in (1), and any positive definite[1] kernel $K : \mathcal{X}^2 \to \mathbb{R}$ induces an embedding $\phi : \mathcal{X} \to \{0,1\}^{\mathcal{X}}$ via

$$\phi(x) \quad = \quad K(x, \cdot)$$

(see Schölkopf and Smola 2002), we shall speak of linear separability by ϕ or by K interchangeably, as dictated by convenience.

An immediate question is whether, under our conventions, every concept class is linearly separable. A construction of the requisite ϕ given any $\mathcal{X}$ and $\mathcal{C}$ would provide a positive answer; an example of $\mathcal{X}$ and $\mathcal{C}$ for which no such embedding exists would resolve the question negatively.

[1] In the sense that for any finite set $\{x_i \in \mathcal{X} : 1 \le i \le m\}$, the $m \times m$ matrix $G = (g_{ij})$ given by $g_{ij} = K(x_i, x_j)$ is always positive definite.

4. Every Concept Class is Linearly Separable

The question raised in Section 3 turns out to have an affirmative answer, as we prove in the following theorem.

Theorem 1 *Every countable concept class $\mathcal{C}$ over a countable instance space $\mathcal{X}$ is finitely linearly separable.*

Proof We present a constructive proof, by describing an explicit mapping $\phi : \mathcal{X} \to \{0,1\}^{\mathbb{N}}$ under which all concepts $c \in C$ are finitely linearly separable. Recall that both the embedding by concept and embedding by instance, described in the previous section, rendered all concepts linearly separable, but were possibly not finitely supported. Hence, the "trick" in our construction is the introduction of *size functions* that measure the *complexity* of both individual instances and individual concepts, thus providing a notion of capacity control and adaptive regularization, and leading to finite separability.

We thus begin by defining the notion of a *size function* $f : \mathcal{A} \to \mathbb{N}$, mapping any countable set $\mathcal{A}$ to the naturals. We require a size function to have finite level sets:

$$\left| f^{-1}(n) \right| \quad < \quad \infty, \qquad \forall n \in \mathbb{N}.$$

In words, f assigns at most finitely many elements of $\mathcal{A}$ to any fixed size. Any countable $\mathcal{A}$ has such a size function (in fact, there are infinitely many size functions on $\mathcal{A}$). We denote by $|\cdot|$ and $\|\cdot\|$ some fixed choices of size functions on $\mathcal{X}$ and $\mathcal{C}$, respectively.

Recall that our goal is to construct an embedding $\phi : \mathcal{X} \to \{0,1\}^{\mathbb{N}}$ with $\|\phi(x)\|_0 < \infty$ for all $x \in \mathcal{X}$ such that for all $c \in \mathcal{C}$ there is a $w = w(c) \in \mathbb{R}^{\mathbb{N}}$ with $\|w\|_0 < \infty$ such that

$$c = \{x \in \mathcal{X} : \langle w, \phi(x) \rangle > 0\}.$$

We will construct the requisite *canonical* embedding ϕ as the direct product of two auxiliary embeddings, α and β,

$$\phi(x) = \alpha(x) \otimes \beta(x). \tag{2}$$

First, we define the *embedding by instance* $\alpha : \mathcal{X} \to \{0,1\}^{\mathcal{X}}$ by

$$[\alpha(x)]_u \quad = \quad \mathbb{1}_{\{u=x\}}, \qquad u \in \mathcal{X};$$

obviously, $\|\alpha(x)\|_0 = 1$ for all $x \in \mathcal{X}$ (recall our vector-indexing conventions, set down in Section 2). Second, using the size function as notion of complexity, we define a *regularized embedding by concept* $\beta : \mathcal{X} \to \{0,1\}^{\mathcal{C}}$ by

$$[\beta(x)]_c \quad = \quad \mathbb{1}_{\{x \in c\}} \mathbb{1}_{\{\|c\| \le |x|\}}, \qquad c \in \mathcal{C}; \tag{3}$$

since size functions have finite level sets, in contrast to the standard embedding by concept, we have $\|\beta(x)\|_0 < \infty$ for any $x \in \mathcal{X}$.

We now show that the embedding (2) renders all concepts $c \in \mathcal{C}$ finitely linearly separable, by explicitly constructing hyperplane vectors $w^{\alpha} \in \mathbb{R}^{\mathcal{X}}$ and $w^{\beta} \in \mathbb{R}^{\mathcal{C}}$ corresponding to the mappings α and β, respectively. To this end, it is helpful to keep in mind the dual roles of $\mathcal{X}$ and $\mathcal{C}$.

Pick any $\hat{c} \in C$. Since the embedding α involved no regularization, we introduce a complexity restriction in the corresponding hyperplane vector $w^{\alpha} \in \mathbb{R}^{\mathcal{X}}$ as follows

$$[w^{\alpha}]_u \quad = \quad \mathbb{1}_{\{u \in \hat{c}\}} \mathbb{1}_{\{|u| < \|\hat{c}\|\}}, \qquad u \in \mathcal{X};$$

since size functions have finite level sets, we have $\|w^\alpha\|_0 < \infty$. Thus,

$$\begin{aligned}\langle w^\alpha, \alpha(x)\rangle &= \sum_{u\in\mathcal{X}} [w^\alpha]_u [\alpha(x)]_u \\ &= \sum_{u\in\mathcal{X}} \mathbb{1}_{\{u\in\hat{c}\}} \mathbb{1}_{\{|u|<\|\hat{c}\|\}} \mathbb{1}_{\{x=u\}} \\ &= \mathbb{1}_{\{x\in\hat{c}\}} \mathbb{1}_{\{|x|<\|\hat{c}\|\}}. \end{aligned} \tag{4}$$

For the second embedding β, no further regularization is needed, and we define its corresponding hyperplane vector $w^\beta \in \mathbb{R}^{\mathcal{C}}$ by

$$[w^\beta]_c = \mathbb{1}_{\{c=\hat{c}\}}, \qquad c \in \mathcal{C};$$

note that $\left\|w^\beta\right\|_0 = 1$. Now

$$\begin{aligned}\left\langle w^\beta, \beta(x)\right\rangle &= \sum_{c\in\mathcal{C}} [w^\beta]_c [\beta(x)]_c \\ &= \sum_{c\in\mathcal{C}} \mathbb{1}_{\{c=\hat{c}\}} \mathbb{1}_{\{x\in c\}} \mathbb{1}_{\{\|c\|\le|x|\}} \\ &= \mathbb{1}_{\{x\in\hat{c}\}} \mathbb{1}_{\{|x|\ge\|\hat{c}\|\}}. \end{aligned} \tag{5}$$

Since ϕ is the direct product of the two embeddings α and β, the corresponding hyperplane is the direct product of the two hyperplanes:

$$w = w^\alpha \otimes w^\beta.$$

Note that by construction, both ϕ and w are finite:

$$\|\phi(x)\|_0 = \|\alpha(x)\|_0 + \|\beta(x)\|_0 < \infty \quad \text{and} \quad \|w\|_0 = \|w^\alpha\|_0 + \|w^\beta\|_0 < \infty.$$

Combining (4) and (5), we get

$$\begin{aligned}\langle w, \phi(x)\rangle &= \langle w^\alpha, \alpha(x)\rangle + \left\langle w^\beta, \beta(x)\right\rangle \\ &= \mathbb{1}_{\{x\in\hat{c}\}} \mathbb{1}_{\{|x|<\|\hat{c}\|\}} + \mathbb{1}_{\{x\in\hat{c}\}} \mathbb{1}_{\{|x|\ge\|\hat{c}\|\}} \\ &= \mathbb{1}_{\{x\in\hat{c}\}}\end{aligned}$$

which shows that w is indeed a linear separator (with finite support) for $\hat{c}$. ■

Next, we observe that linearly separable concepts may be described by finitely many instances, via the following "representer theorem". Note that this result holds for *any* separating embedding—not just the canonical one constructed in Theorem 1.

Theorem 2 *If a concept class $\mathcal{C}$ over an instance space $\mathcal{X}$ is linearly separable under ϕ (equivalently, under the induced kernel $K(x,y) = \langle\phi(x), \phi(y)\rangle$), there are $s_i \in \mathcal{X}$ and $\alpha_i \in \mathbb{R}$, $i = 1, \ldots, m$, such that*

$$c = \left\{ x \in \mathcal{X} : \sum_{i=1}^m \alpha_i K(s_i, x) > 0 \right\}.$$

Proof Suppose a concept $c \in \mathcal{C}$ is linearly separable with vector w, that is, $c = \{x \in \mathcal{X} : \langle w, \phi(x) \rangle > 0\}$. It follows that c is also separable under the finite dimensional embedding $\phi_D : \mathcal{X} \to \{0,1\}^D$ where $D = \max\{i : w_i \neq 0\}$ and ϕ_D is the truncation of ϕ by D:

$$[\phi_D(x)]_i \quad = \quad [\phi(x)]_i, \qquad i = 1, \ldots, D.$$

Define the equivalence relation $\equiv_D$ on $\mathcal{X}$ via

$$x \equiv_D y \quad \Leftrightarrow \quad \phi_D(x) = \phi_D(y)$$

and notice that $\equiv_D$ partitions $\mathcal{X}$ into finitely many equivalence classes. Let $x_1, x_2, \ldots, x_m$ be chosen arbitrarily, one from each equivalence class, and define $y_i := \chi_c(x_i)$. Let w' be the maximum-margin hyperplane for the labeled sample $(x_i, y_i)_{i=1}^m$ (see (9) for the definition of margin). This construction ensures that w' is a linear separator for c. By the representer theorem of Kimeldorf and Wahba (1971), it follows that w' admits a representation of the form

$$w' = \sum_{i=1}^m \alpha_i \phi(x_i),$$

and the claim follows. ■

A few remarks regarding Theorem 1 are in order. First, note that the construction of the canonical embedding depends on the choice of the size functions $|\cdot|$ and $\|\cdot\|$ on $\mathcal{X}$ and $\mathcal{C}$. These size functions induce a natural notion of complexity for instances $x \in \mathcal{X}$ and concepts $c \in \mathcal{C}$. Hence, our construction has analogies to various settings where classifier complexity can adaptively grow with the sample, such as structural risk minimization (SRM) in machine learning (Vapnik, 1982), and minimum description length (MDL) and other information theoretic criteria in statistics (Rissanen, 1989). We shall have more to say about SRM in particular, below.

Observe that any nondecreasing transformation $f : \mathbb{N} \to \mathbb{N}$ of a size function $|\cdot|$ produces another valid size function $|x|' := f(|x|)$. These affect the embedding in the following way. Consider a canonical embedding ϕ induced by $(|\cdot|, \|\cdot\|)$, and consider a transformed embedding ϕ' induced by $(f(|\cdot|), \|\cdot\|)$. If f is a very rapidly growing function then the truncation in (3) becomes ineffective, and in the limit of $f(\cdot) \equiv \infty$, ϕ' essentially becomes a pure embedding by concept. In the other extreme of a very slowly increasing f, the truncation in (3) becomes too restrictive. In the limit of $f(\cdot) \equiv \text{const}$, ϕ' effectively becomes an embedding by instance.

A final note is that the separability result does not preclude the trivial scenario where the only coordinate separating a concept is the concept itself. To parse the last statement, recall that Theorem 1 states that under the canonical embedding, for each concept $c \in \mathcal{C}$ there is a finite $N = N(c)$ such that c is separable in $\{0,1\}^N$. However, it may be the case that the separator w is trivial, in the sense that $w_i \equiv \mathbb{1}_{\{i=c\}}$. In other words, the separability result implies that for each $c \in \mathcal{C}$ there is a finite collection of $\{c_i \in \mathcal{C} : \|c_i\| \le \|c\|, 1 \le i \le m\}$ with coefficients $\alpha_i \in \mathbb{R}$ such that

$$c \quad = \quad \sum_{i=1}^m \alpha_i c_i, \tag{6}$$

where the relation above is symbolic shorthand for the statement that

$$c \quad = \quad \{x \in \mathcal{X} : \sum_{i=1}^m \alpha_i \mathbb{1}_{\{x \in c_i\}} > 0\}.$$

When a relation of type (6) holds, we say that c is *linearly decomposable* into the concepts $\{c_i\}$. No claims of uniqueness are made, and certainly any concept c is trivially decomposable into the singleton $\{c\}$. However, in many situations, concepts indeed decompose. As a simple example, define the concept c_{10} to be the set of all binary numbers, written as strings in $\{0,1\}^*$, divisible by ten, and define c_1, c_2 and c_5 analogously. Then it is straightforward to verify that

$$c_{10} = c_2 + c_5 - c_1,$$

in the sense of (6), since a number is divisible by 10 iff it is divisible by both 2 and 5.

As far as implications for learnability, Theorem 1 is most relevant when the target concept is linearly decomposable into a *small* collection of *simpler* concepts, as illustrated in the last example. In such cases, our approach can offer a substantial advantage over existing methods.

We may formalize the latter observation as a "universal learnability" theorem:

Theorem 3 *Let $\mathcal{C}$ be a concept class over $\mathcal{X}$, both countable, with size functions $\|\cdot\|$ and $|\cdot|$, respectively. Define the family of kernels:*

$$K_n(x,y) \quad = \sum_{c:\|c\| \le n} \mathbb{1}_{\{x \in c\}} \mathbb{1}_{\{y \in c\}} \tag{7}$$

for $n \in \mathbb{N}$. Then, given an oracle for computing K_n, we can efficiently learn $\mathcal{C}$ from labeled examples.

More explicitly, let $c \in \mathcal{C}$ be a fixed concept, and let S be a sequence of m instances independently sampled from an arbitrary distribution P, and labeled according to c.

There is an efficient algorithm `UnivLearn`*, which inputs a labeled sample and outputs a classifier $\hat{f} : \mathcal{X} \to \{-1,1\}$ which, with probability at least $1-\delta$, achieves a small generalization error:*

$$P\{\hat{f}(x) \neq \chi_c(x)\} \quad \le \quad \frac{2}{m}\left(R(c)\log(8em)\log(32m) + \log(8m/\delta)\right), \quad \text{for } m \ge \theta^{-1}(\|c\|), \tag{8}$$

where $R(c)$ does not depend on the distribution P and $\theta, \theta^{-1} : \mathbb{N} \to \mathbb{N}$ are fixed monotone increasing functions with polynomial growth.

We defer the proof of Theorem 3 to Section 5, as it requires some preliminary definitions and results. The regularization function θ is chosen by the learner and represents his desired tradeoff between computational complexity and generalization error; this is further elucidated in Remark 7. We stress that the size of the target concept, $\|c\|$, is unknown to the learner—otherwise regularization would not be necessary. Finally, Remarks 4 and 5 below point out the non-trivial nature of Theorem 3; we are not aware of any way to obtain it as an immediate corollary of known results.

Remark 4 *We emphasize that the bound in Theorem 3 is distribution-free and recall that a distribution-dependent "learning" bound is trivial to obtain. Define $\hat{f}$ to be the rote memorizer: $\hat{f}(x) = y(x)$ if the example x appears in the sample S with label y and $\hat{f}(x) = -1$ otherwise. In this case, we have*

$$P\{\hat{f}(x) \neq \chi_c(x)\} \le P(x \notin S) = P(\mathcal{X} \setminus S),$$

and the latter quantity clearly tends to 0 as $m \to \infty$. All this bound says is that when we have observed a large enough portion of the world, we can be sure that most test examples will simply be recycled training examples. However, for any concept containing infinitely many instances, one can find an adversarial distribution P so that $P(\mathcal{X} \setminus S)$ will tend to zero arbitrarily slowly.

Remark 5 *The dependence of the required number of samples on the unknown concept c in the generalization bound (8) is inevitable whenever $\mathcal{C}$ has infinite VC-dimension. As a simple example, note that to learn an n-state DFA, one needs to observe at least n strings (at least one string ending at each state).*

5. Margin Bounds

In this section, we recall some classic margin bounds and their implications for learnability. Let $\mathcal{C}$ be a concept class over the instance space $\mathcal{X}$. Suppose there is some family of embeddings $\phi_d : \mathcal{X} \to \{0,1\}^{N_d}$; at this point, we assume nothing about the relationship between $\{\phi_d\}$ and $\mathcal{C}$. Let $S \subset \mathcal{X}$ be a finite sample with labels $Y \in \{-1,1\}^S$ consistent with some $c \in \mathcal{C}$, that is, $Y_x = \chi_c(x) = 2\mathbb{1}_{\{x \in c\}} - 1$ for all $x \in S$.

We define the *sample margin* of (S,Y) under ϕ_d as follows:

$$\hat{\gamma}_d(S,Y) \quad := \sup_{w \in B(N_d)} \min_{x \in S} Y_x \langle w, \phi_d(x) \rangle \tag{9}$$

where $B(k) := \{x \in \mathbb{R}^k : \|x\|_2 \le 1\}$. The *sample radius* of S under ϕ_d is defined to be

$$\hat{\rho}_d(S) \quad := \quad \max_{x \in S} \|\phi_d(x)\|_2 .$$

Analogously, define the *intrinsic margin* of c under ϕ_d to be

$$\gamma_d(c) \quad = \sup_{w \in B(N_d)} \inf_{x \in \mathcal{X}} \chi_c(x) \langle w, \phi_d(x) \rangle$$

and the *radius* of ϕ_d to be

$$\rho_d \quad = \quad \sup_{x \in \mathcal{X}} \|\phi_d(x)\|_2 .$$

The case where γ_d or $\hat{\gamma}_d$ is negative is not excluded; it arises when ϕ_d fails to separate the given concept or sample. Note that since for $A \subset B$, we have $\inf_A f \ge \inf_B f$ and $\sup_A f \le \sup_B f$, the relation $S \subset \mathcal{X}$ implies that

$$\hat{\gamma}_d(S,Y) \ge \gamma_d(c) \quad \text{and} \quad \hat{\rho}_d(S) \le \rho_d . \tag{10}$$

The following theorem is a slight rewording of Shawe-Taylor et al. (1998, Theorem 4.5):

Theorem 6 (Shawe-Taylor et al. 1998) *Suppose m instances are drawn independently from a distribution whose support is contained in a ball in $\mathbb{R}^n$ centered at the origin, of radius ρ. If we succeed in correctly classifying all of them by a hyperplane through the origin with margin γ, then with confidence $1-\delta$ the generalization error will be bounded from above by*

$$\frac{2}{m}\left(R \log(8em/R) \log(32m) + \log(8m/\delta)\right), \tag{11}$$

where $R = R(\rho,\gamma) = \lfloor 577\rho^2/\gamma^2 \rfloor$.

We are now in a position to prove the "universal learnability" theorem:

Proof [of Theorem 3] Recall that (S,Y) is our finite labeled sample consistent with some unknown target concept c. For each $n \in \mathbb{N}$, let $\phi_n : \mathcal{X} \to \{0,1\}^{N_n}$ be the embedding associated with K_n, defined in (7), where

$$N_n = |\mathcal{C}_n| = |\{c \in \mathcal{C} : \|c\| \le n\}|.$$

Let the margins $\gamma_n(c)$, $\hat{\gamma}_n(S,Y)$ and radii ρ_n, $\hat{\rho}_n(S)$ be as defined above. Define the *normalized margins*

$$\Delta_n(c) := \frac{\gamma_n(c)}{\rho_n}, \quad \text{and} \quad \hat{\Delta}_n(S,Y) := \frac{\hat{\gamma}_n(S,Y)}{\hat{\rho}_n(S)}.$$

Additionally, define $n_0 = n_0(c)$ to be the smallest n for which K_n linearly separates c:

$$n_0(c) := \min\{n \in \mathbb{N} : \gamma_n(c) > 0\} \tag{12}$$

and define

$$\Delta_0(c) := \Delta_{n_0(c)}(c).$$

The algorithm `UnivLearn` requires some fixed monotone increasing function $\theta : \mathbb{N} \to \mathbb{N}$, as mentioned in the statement of the theorem (for a concrete choice, one may take $\theta(m) = \lceil m^{1/2} \rceil$). Recall the theorem's condition that $m \ge \theta^{-1}(\|c\|)$ which (by Theorem 1) implies that $K_{\theta(m)}$ indeed separates the unknown concept c. The algorithm proceeds by computing the normalized sample margins $\hat{\Delta}_n(S,Y)$ for $n = 1, \ldots, \theta(m)$ and sets

$$\hat{\Delta}_*(S,Y) := \max\{\hat{\Delta}_n(S,Y) : 1 \le n \le \theta(m),\ \hat{\gamma}_n(S,Y) > 0\}$$

with corresponding $n_* \in \{1, \ldots, \theta(m)\}$. The condition $m \ge \theta^{-1}(\|c\|)$ ensures that the latter is well defined.

By (10) we have that $\hat{\Delta}_n(S,Y) \ge \Delta_n(c)$ for all $\{n \in \mathbb{N} : \gamma_n(c) > 0\}$. In addition, the condition $m \ge \theta^{-1}(\|c\|)$ combined with Theorem 1 ensure that

$$n_0(c) \le \|c\| \le \theta(m).$$

Therefore, we have

$$\hat{\Delta}_*(S,Y) \ge \Delta_0(c). \tag{13}$$

The classifier $\hat{f} : \mathcal{X} \to \{-1,1\}$ which `UnivLearn` outputs is the maximum-margin hyperplane under the embedding ϕ_{n_*}. It follows from Theorem 6 that the bound in (8) holds with $R(c) = \lfloor 577/\Delta_0(c)^2 \rfloor$. Note that in (8) we have opted for a slightly coarser but simpler bounds.

■

Remark 7 *To explain the role of the function $\theta(m)$, let us attempt to simplify the algorithm* `UnivLearn`*. Recall the definition of n_0 in (12), and define $\hat{n}_0$ to be the first n for which we perfectly classify the training set, that is, $\hat{n}_0 = \min_n\{\hat{\gamma}_n(S,Y) > 0\}$ and let $\hat{\Delta}_0(S,Y) := \hat{\Delta}_{\hat{n}_0}(S,Y)$. If it were the case that*

$$\hat{\Delta}_0(S,Y) \ge \Delta_0(c), \tag{14}$$

then the maximum-margin classifier $\hat{f}$ corresponding to $\phi_{\hat{n}_0}$ would satisfy the desired bound in (8). Unfortunately, the assertion in (14) is generally false, since it is possible for a small sample to be separable under K_n with $n < n_0$ and margin $\hat{\gamma}_n(S,Y) \ll \gamma_{n_0}(c)$. Because of this potentially very small margin, stopping prematurely at this n would render our theorem false. There seems to be no way around requiring the learner to try every $n \in \mathbb{N}$. To make the algorithm terminate in finite time for each given sample, we cut off the search at $n = \theta(m)$. Note that the theorem is only claimed to hold for sufficiently large sample sizes (namely, $m \geq \theta^{-1}(\|c\|)$)—but this is a feature of many bounds, especially where the complexity of the target concept is unknown (cf. Theorem 8). Since θ is a monotonically increasing function, we are guaranteed that eventually the sample size m will attain $\theta^{-1}(\|c\|)$, which is what the claim in (13) hinges upon. If θ grows too quickly (say, $\theta(\cdot) = \Omega(\exp(\cdot))$), the algorithm will take too long to run. If θ grows too slowly (say, $\theta(\cdot) = O(\log(\cdot))$), the generalization bound will only hold for huge sample sizes. The suggested rate of $\theta(m) = \lceil m^{1/2} \rceil$) seems a reasonable compromise.

Theorem 3 has a natural interpretation in the structural risk minimization (SRM) framework (Vapnik, 1982). Let us quote a well-known result, as it appears in Shawe-Taylor et al. (1996); for a detailed discussion of VC-dimension, see Vapnik's books (Vapnik, 1982, 1998) or any standard reference on learning theory such as Kearns and Vazirani (1997).

Theorem 8 (Shawe-Taylor et al. 1996) *Let H_i, $i = 1,2,\ldots$ be a sequence of hypothesis classes mapping $\mathcal{X}$ to $\{0,1\}$ such that $\mathrm{VCdim}(H_i) = i$ and let P be a probability distribution on $\mathcal{X}$. Let p_d be any set of positive numbers satisfying $\sum_{d=1}^{\infty} p_d = 1$. With probability $1-\delta$ over m independent examples drawn according to P, for any d for which the learner finds a consistent hypothesis h in H_d, the generalization error of h is bounded above by*

$$\varepsilon(m,d,\delta) = \frac{4}{m}\left(d\log\left(\frac{2em}{d}\right) + \log\left(\frac{1}{p_d}\right) + \log\left(\frac{4}{\delta}\right)\right),$$

provided $m \geq d$.

As a concrete choice of p_d, one may always take $p_d = 2^{-d}$. Viewing Theorems 3 and 8 through a computational lens, the two approaches may be contrasted by their "computational primitives". The SRM approach requires one to find a consistent hypothesis in a given complexity class (if one exists). Whenever the latter problem is efficiently solvable, SRM is quite satisfactory as a learning paradigm. In many interesting cases, however, the latter problem is intractable—as is the case for DFAs, for example (see Introduction). Our approach in Theorem 3 is to consider a different computational primitive—namely, the generic kernel

$$K_n(x,y) = |\{c \in \mathcal{C} : \|c\| \leq n, x \in c, y \in c\}|.$$

The margin-based bound may be significantly sharper than the generic SRM bound since it is sensitive to the target concept and its intrinsic margin. Additionally, the SRM bound is not directly applicable to our setting since the VC-dimension of linear combinations of functions may grow linearly with the number of terms (Anthony and Bartlett, 1999). The roles of θ in Theorem 3 and $\{p_d\}$ in Theorem 8 are analogous in that both encode a prior belief regarding hypothesis complexity.

The performance of `UnivLearn` is readily quantified:

Corollary 9 *If a labeled sample of size m is consistent with some concept $c \in \mathcal{C}$, and there is an oracle for computing $K_n(x,y)$ in time $O(1)$, then there is an algorithm with running time $O(m^2\theta(m))$ that achieves, with probability $1-\delta$, a generalization error of at most*

$$\frac{2}{m}\left(R(c)\log(8em)\log(32m)+\log(8m/\delta)\right),$$

for $m \geq \theta^{-1}(\|c\|)$.

Proof Using the oracle, the nth Gram matrix, defined by $(G_n)_{ij} = K_n(x_i,x_j)$, for $x_i,x_j \in S$, is computable in time $O(m^2)$. For a given Gram matrix, the margin may be computed in time $O(m)$ by a primal algorithm such as Pegasos (Shalev-Shwartz et al., 2007). The different Gram matrices and margins are computed $\theta(m)$ times. ■

Of course, at this point we have simply traded the generally intractable problem of finding a consistent $h \in H_d$ for the problem of evaluating $K_n(x,y)$. The latter is unlikely to have an efficient solution in general, and may well be intractable in many cases of interest. However, as we shall see below, under natural assumptions K_n admits an efficient stochastic approximation $\tilde{K}_n$ and Corollary 9 has a suitable extension in terms of $\tilde{K}_n$.

6. Approximating K_n

In the previous section we showed that the generic kernel family K_n renders all concepts $c \in \mathcal{C}$ linearly separable. However, exact evaluation of this kernel involves the summation in (7), which may contain a super-exponential number of terms. For example, a natural size function on DFAs is the number of states. Since the VC-dimension of n-state DFAs is $\Theta(n\log n)$ (Ishigami and Tani, 1997), there are $2^{\Theta(n\log n)}$ such concepts. Hence, a direct brute-force evaluation of the corresponding K_n is out of the question. Though we consider the complexity of computing K_n for DFAs to be a likely candidate for #P-complete, we have no proof of this; there is also the possibility that some symmetry in the problem will enable a clever efficient computation.

Instead, we propose an efficient randomized approximation to the generic K_n. All we require are oracles for computing $|\mathcal{C}_n|$ and for uniform sampling from $\mathcal{C}_n$, where

$$\mathcal{C}_n := \{c \in \mathcal{C} : \|c\| \leq n\}: \tag{15}$$

Assumption 10

(i) *There is a sampling algorithm with running time $T_{\text{SAM}}(\mathcal{C},n) = O(\text{poly}(n))$ that outputs random concepts $c \in \mathcal{C}_n$, each with probability $|\mathcal{C}_n|^{-1}$. Additionally, k distinct concepts may be drawn (i.e., without replacement) in expected time $O(kT_{\text{SAM}}(\mathcal{C},n))$.*

(ii) *There is an algorithm that computes $|\mathcal{C}_n|$, in time $O(1)$.*

We make an additional assumption regarding the time complexity of instance/concept membership queries:

Assumption 11 *For each $x \in \mathcal{X}$ and $c \in \mathcal{C}$, the characteristic function $\chi_c(x) = 2\mathbb{1}_{\{x\in c\}} - 1$ is computable in time $O(|x|)$.*

Note that sampling without replacement is a mild condition. Collisions are likely to occur when sampling from small sets, in which case one could simply enumerate the elements of $\mathcal{C}_n$. When the latter is large, however, uniform sampling is rather unlikely to generate collisions, and various hashing schemes can be employed to ensure this in practice.

For the purpose of approximating the kernel, it is convenient to normalize it as follows:

$$\bar{K}_n(x,y) := |\mathcal{C}_n|^{-1} K_n(x,y), \tag{16}$$

Note that the margin bounds of Section 5 are invariant under rescaling of the kernel by any constant.

Having stated our basic assumptions, we are ready to prove some results. The first one deals with efficient approximation of the generic kernel:

Theorem 12 *Let $S \subset \mathcal{X}$ be a sample of size m and the normalized generic kernel $\bar{K}_n$ be defined as in (16). Then, for any $0 < \varepsilon, \delta < 1$, there is a randomized algorithm with deterministic running time*

$$T \quad = \quad O\left(\frac{1}{\varepsilon^2} T_{\mathrm{SAM}}(\mathcal{C}, n) \log\left(\frac{m}{\delta}\right)\right),$$

that produces a (random) kernel $\tilde{K}_n$ with the following properties

(a) *$\tilde{K}_n$ is a positive semidefinite function on $\mathcal{X}^2$*

(b)

$$\mathbf{E}[\tilde{K}_n(x,y)] \quad = \quad \bar{K}_n(x,y)$$

for all $x, y \in \mathcal{X}$

(c) *with probability at least $1-\delta$,*

$$\left|\bar{K}_n(x,y) - \tilde{K}_n(x,y)\right| \quad \leq \quad \varepsilon,$$

for all $x, y \in S$,

(d) *for all $x, y \in \mathcal{X}$, $\tilde{K}_n(x,y)$ is computable in deterministic time*

$$O\left(\frac{1}{\varepsilon^2} \log\left(\frac{m}{\delta}\right) (|x| + |y|)\right).$$

Proof The approximation algorithm amounts to sampling T concepts, $\{c_i : 1 \leq i \leq T\}$ uniformly at random (without replacement) from $\mathcal{C}_n$, where Assumption 10 ensures that the latter sampling scheme is computationally feasible.

The random concepts $\{c_i : 1 \leq i \leq T\}$ are used to define the random kernel $\tilde{K}_n$:

$$\tilde{K}_n(x,y) \quad = \quad \frac{1}{T} \sum_{i=1}^{T} \mathbb{1}_{\{x \in c_i\}} \mathbb{1}_{\{y \in c_i\}}, \qquad x, y \in \mathcal{X}. \tag{17}$$

Since $\tilde{K}_n$ is constructed as a (semi) inner product, it is non-negative but still not strictly positive definite, as it may fail to separate distinct points. Hence, (a) is proved.

To prove (b), fix $x,y \in \mathcal{X}$ and define $D_{xy} \subset \mathcal{C}_n$ by

$$D_{xy} \quad = \quad \{c \in \mathcal{C}_n : x,y \in c\};$$

this is the set of those $c \in \mathcal{C}$ with $\|c\| \leq n$ which contain both x and y. Since any c drawn uniformly from $\mathcal{C}_n$ contains both x and y with probability $|D_{xy}| \,/\, |\mathcal{C}_n|$ it follows that the approximate kernel

$$\tilde{K}_n(x,y) = \frac{1}{T} \sum_{i=1}^{T} \mathbb{1}_{\left\{c_i \in D_{xy}\right\}} \tag{18}$$

is an unbiased estimator for

$$\bar{K}_n(x,y) \quad = \quad |\mathcal{C}_n|^{-1} \sum_{c \in \mathcal{C}_n} \mathbb{1}_{\left\{c \in D_{xy}\right\}} \quad = \quad \frac{|D_{xy}|}{|\mathcal{C}_n|};$$

this proves (b).

Our arguments show that the random variable $\mathbb{1}_{\left\{c_i \in D_{xy}\right\}}$ has a Bernoulli distribution with mean $\bar{K}_n(x,y)$. Hence, applying Hoeffding's bound (Hoeffding, 1963), which also holds for variables sampled without replacement, to (18) yields

$$\mathbf{P}\left\{\left|\tilde{K}_n(x,y) - \bar{K}_n(x,y)\right| > \varepsilon\right\} \quad \leq \quad 2\exp(-2T\varepsilon^2). \tag{19}$$

The latter can be inverted to make the claim in (c) hold for a given x,y whenever $T > (2\varepsilon^2)^{-1}\log(2/\delta)$. Applying the union bound to (19), we have established (c), with

$$T \quad = \quad \frac{1}{2\varepsilon^2} \log\left(\frac{m^2+m}{\delta}\right).$$

Finally, (d) holds by Assumptions 10 and 11. ■

Our next observation is that perturbing a kernel pointwise by a small additive error preserves linear separability:

Theorem 13 *Let $K : \mathcal{X}^2 \to \mathbb{R}$ be a kernel, and $S \subset \mathcal{X}$ be a finite sample with labels $Y \in \{-1,1\}^S$. Assume that (S,Y) is separable under the kernel K with margin γ, that is,*

$$\sup_{w \in \mathbb{R}^{\mathbb{N}}} \min_{s \in S} \frac{Y_s \langle w, \phi(s) \rangle}{\|w\|_2} \quad \geq \quad \gamma > 0,$$

where ϕ is the embedding associated with K. Let K' be another kernel, which is uniformly close to K on S:

$$\left|K(s,t) - K'(s,t)\right| \leq \varepsilon \qquad\qquad \forall s,t \in S. \tag{20}$$

Then, for $\varepsilon < \gamma^2$, the sample (S,Y) is also separable under K', with a positive margin.

Proof Let $f(x)$ be the maximum-margin hyperplane classifier corresponding to the training set S, under the kernel K. By the representer theorem, it can written as

$$f(x) = \langle \phi(x), w\rangle = \sum_{s\in S} \alpha_s K(x,s).$$

Since (S,Y) is linearly separable, $f(x)$ satisfies the constraint

$$Y_s f(s) \geq 1 \qquad \forall s \in S.$$

By the KKT conditions (Schölkopf and Smola, 2002), its margin is given by

$$\gamma^{-2} \;\;=\;\; \sum_s |\alpha_s|.$$

We now consider the following linear classifier, obtained by replacing K by K':

$$f'(x) = \sum_{s\in S} \alpha_s K'(x,s).$$

By (20), for each $x,s \in S$,

$$K'(x,s) = K(x,s) + \varepsilon\eta_{x,s}$$

where $|\eta_{x,s}| \leq 1$. Hence, for all $x \in S$,

$$f'(x) = f(x) + \varepsilon \sum_{s\in S} \eta_{x,s}\alpha_s.$$

Therefore,

$$Y_s f'(s) \geq 1 - \varepsilon \sum_s |\alpha_s| = 1 - \frac{\varepsilon}{\gamma^2},$$

which shows that f' is a linear separator for (S,Y) as long as $\varepsilon < \gamma^2$. ■

The next result quantifies the amount by which the margin may shrink when an approximate kernel is constructed using a random subset of the features:

Theorem 14 *Let a labeled sample (S,Y) be given, with $S \subset \mathcal{X}$. Let the kernel K be induced by the embedding $\phi : \mathcal{X} \to \{0,1\}^F$, where F is a finite feature set, as follows:*

$$K(s,t) := \frac{1}{|F|}\langle \phi(s), \phi(t)\rangle.$$

Consider a feature subset $F' \subset F$. Define $\phi' : \mathcal{X} \to \{0,1\}^{F'}$ to be the restriction of ϕ to $\{0,1\}^{F'}$ and K' to be corresponding kernel:

$$K'(s,t) := \frac{1}{|F'|}\left\langle \phi'(s), \phi'(t)\right\rangle.$$

Suppose that (S,Y) is linearly separable under K with margin $\gamma > 0$. If K' is ε-close to K on S in the sense of (20) with $\varepsilon < \gamma^2$, then (S,Y) is linearly separable under the kernel K', with margin γ' bounded from below by

$$\gamma' \geq \frac{|F'|}{|F|}\left(\gamma - \frac{\varepsilon}{\gamma}\right).$$

Proof Let $f : \mathcal{X} \to \mathbb{R}$ be the maximum-margin separator under K:

$$f(x) = \sum_{s \in S} \alpha_s K(x,s) = \frac{1}{|F|} \sum_{s \in S} \alpha_s \langle \phi(x), \phi(s) \rangle = \langle w, \phi(x) \rangle .$$

Let us define f' as f with K replaced by K':

$$f'(x) = \sum_{s \in S} \alpha_s K'(x,s) = \frac{1}{|F'|} \sum_{s \in S} \alpha_s \left\langle \phi'(x), \phi'(s) \right\rangle .$$

By Theorem 13 we have that f' is a linear separator for (S,Y), satisfying

$$Y_s f'(s) \geq 1 - \varepsilon \sum_s \alpha_s = 1 - \frac{\varepsilon}{\gamma^2}, \qquad s \in S.$$

Observe that

$$f'(x) = \frac{|F|}{|F'|} \left\langle w|_{F'}, \phi'(x) \right\rangle$$

where $w|_{F'}$ is the restriction of w to $\mathbb{R}^{F'}$. Thus, letting

$$w' := \frac{|F|}{|F'|} \left(1 - \frac{\varepsilon}{\gamma^2} \right)^{-1} w|_{F'},$$

we have that

$$f''(x) := \left\langle w', \phi'(x) \right\rangle$$

linearly separates S and satisfies and $Y_s f''(s) \geq 1$ for $s \in S$. Since the margin attained by the hyperplane w is given by $1/\|w\|$ (Schölkopf and Smola, 2002), the claim follows. ■

Our final result for this section—and perhaps the highlight of the paper—is the following "approximate" version of Corollary 9:

Theorem 15 *Let $\mathcal{C}$ be a concept class over $\mathcal{X}$. Let $S \subset \mathcal{X}$ be a sequence of m instances sampled independently from an arbitrary distribution P, and labeled according to some unknown $c \in \mathcal{C}$. For any $\delta > 0$, there is a randomized algorithm* `ApproxUnivLearn`*, which outputs a classifier $\hat{f} : \mathcal{X} \to \{-1,1\}$ such that with probability at least $1-\delta$ it achieves a small generalization error,*

$$P\{\hat{f}(x) \neq \chi_c(x)\} \quad \leq \quad \frac{2}{m} \left(4R(c) \log(8em) \log(32m) + \log(16m/\delta) \right) \tag{21}$$

for

$$m \geq \max\{\theta^{-1}(\|c\|), D(c)/2, R(c)^2/8.4 \times 10^4\}, \tag{22}$$

where $0 < D(c), R(c) < \infty$ are constants that depend only on c, and $\theta : \mathbb{N} \to \mathbb{N}$ is discussed in Remark 7.

Furthermore, `ApproxUnivLearn` *has deterministic polynomial complexity, with running time*

$$O\left(m\theta(m) \log(m/\delta) (\ell_{\max} m^2 + T_{\text{SAM}}(\mathcal{C}, \theta(m))) \right),$$

where $\ell_{\max} := \max\{|s| : s \in S\}$).

Proof Let (S,Y) denote a training set consisting of m samples, labeled consistently with the target concept $c \in \mathcal{C}$. We define

$$\begin{aligned} n_0 &:= \min\{n \in \mathbb{N} : \gamma_n(c) > 0\}, \\ \gamma_0(c) &:= \gamma_{n_0}(c), \end{aligned}$$

and

$$\begin{aligned} R(c) &:= \left\lfloor 577/\gamma_0(c)^2 \right\rfloor, \\ D(c) &:= |\mathcal{C}_{n_0}|, \end{aligned}$$

where $\mathcal{C}_n$ is defined in (15) as the set of concepts with complexity at most n.

Given the training set (S,Y), our algorithm constructs $\theta(m)$ different classifiers. According to Theorem 12, each of these classifiers is constructed as follows: For $1 \leq n \leq \theta(m)$, we randomly sample T features, construct the respective approximate kernel, and calculate its resulting max-margin hyperplane, with sample margin $\tilde{\gamma}_n(S,Y)$. Finally, our algorithm chooses the kernel with the largest empirical margin

$$\tilde{\gamma}_* := \max\{\tilde{\gamma}_n(S,Y) : 1 \leq n \leq \theta(m), \tilde{\gamma}_n(S,Y) > 0\}. \tag{23}$$

If the latter set is empty, we leave $\tilde{\gamma}_*$ undefined (however, our analysis below will show that under the conditions of the theorem, this is a highly unlikely event).

To prove the theorem, we need to show that with high probability,

$$\tilde{\gamma}_* \geq \text{Const} \times \gamma_0(c). \tag{24}$$

If this equation holds (whp), then the standard margin bound (11) proves the theorem. There are two ways in which the theorem's generalization bound (21) may fail to hold. The first is due to a particularly unlucky sample whereas the second is due to a bad kernel approximation so that (24) does not hold. Hence, we split the confidence parameter δ so that each of these kinds of failure occurs with probability at most $\delta/2$.

The first step is to note that by definition $\tilde{\gamma}_* \geq \tilde{\gamma}_{n_0}$. The next step is to lower bound the approximate sample margin, constructed with a random approximate kernel, in terms of the true sample margin corresponding to the exact kernel. Such a bound is provided by Theorem 14:

$$\tilde{\gamma}_{n_0}(S,Y) \geq \frac{\min\{T, D(c)\}}{D(c)} \left(\hat{\gamma}_{n_0}(S,Y) - \frac{\varepsilon}{\hat{\gamma}_{n_0}(S,Y)} \right).$$

Note that by condition (22) we have

$$m \quad \geq \quad 4/\gamma_0(c)^4 \geq 4/\hat{\gamma}_{n_0}(S,Y)^4, \tag{25}$$

since $\hat{\gamma}_{n_0}(S,Y) \geq \gamma_0(c)$, as in (10).

Furthermore, in constructing the approximate kernels, we set the precision to be $\varepsilon = 1/\sqrt{m}$. This condition, combined with (25) gives

$$\tilde{\gamma}_* \geq \frac{1}{2} \frac{\min\{T, D(c)\}}{D(c)} \hat{\gamma}_{n_0}(S,Y) \geq \frac{1}{2} \frac{\min\{T, D(c)\}}{D(c)} \gamma_0(c).$$

The last step is to bound the factor multiplying $\gamma_0(c)$ in the last equation. To this end, recall that according to Theorem 12, under the condition $\varepsilon = 1/\sqrt{m}$, the number of random concepts T drawn from $\mathcal{C}_n$ is

$$T = \frac{1}{2\varepsilon^2}\log\left(\frac{m^2+m}{\delta/2}\right) = \frac{m}{2}\log\left(\frac{m^2+m}{\delta/2}\right) > m.$$

Combining this with (22), we have

$$T > m \geq D(c)/2.$$

Hence, $\tilde{\gamma}_* \geq \gamma_0(c)/4$, and applying (11) proves the generalization bound.

It remains to analyze the complexity of `ApproxUnivLearn`. The main loop is executed $\theta(m)$ times. To compute each approximate kernel we sample $O(\varepsilon^{-2}\log(m/\delta))$ concepts from $\mathcal{C}_n$, and evaluate the summation in (17) for each $x,y \in S$. Since $\varepsilon = m^{-1/2}$, the nth Gram matrix is computable in time $O\left(m\log(m/\delta)(\ell_{\max}m^2 + T_{\text{SAM}}(\mathcal{C},n))\right)$. For a given Gram matrix, the margin may be computed in time $O(m)$ by a primal algorithm such as Pegasos (Shalev-Shwartz et al., 2007), which gets absorbed into the preceding $O(m^3\theta(m)\log(m))$ bound.

■

Remark 16 *For* $\theta(m) = \lceil m^{1/2}\rceil$, `ApproxUnivLearn` *has a running time of* $O(m^{3.5}\log m)$. *This may be brought down to* $m^{2+\alpha}$ *for any* $\alpha > 0$, *at the expense of increased sample complexity in (21), by choosing* $\varepsilon = m^{-\beta}$ *with* $\beta \ll 1$. *The role of* $\theta(m)$ *in the tradeoff between running time and sample complexity is discussed in Remark 7;* $\varepsilon(m)$, *taken to be* $m^{-1/2}$ *in the proof above, has an analogous role.*

7. Universal Regular Kernel, New Characterization of Regular Languages

Having developed some general tools for learning countable concept classes, we now apply these to regular languages. For the remainder of the paper, we follow standard formal-language terminology. Thus, Σ will denote a fixed finite alphabet and Σ^* is the free monoid over Σ. Its elements are called *strings* (or *words*) and the length of $x \in \Sigma^*$ is denoted by $|x|$. The latter will be our default size function on Σ^*.

Our definition of a Deterministic Finite-state Automaton (DFA) is a slight modification of the standard one. We define the latter to be a tuple $A = (\Sigma, Q, F, \delta)$ where

- Σ is a finite alphabet
- $Q = \{1, 2, \ldots, n\}$ is a finite set of states
- $F \subset Q$ is the set of the accepting states
- $\delta : Q \times \Sigma \to Q$ is the deterministic transition function.

The only difference between our definition and the common one is that we take the initial state q_0 to be fixed at $q_0 = 1$, while most authors allow it to be an arbitrary $q_0 \in Q$. We write $L(A)$ for the language accepted by the automaton A, and denote the number of states in A by $\mathrm{size}(A)$ or $\|A\|$, interchangeably.

Recall that a language $L \subseteq \Sigma^*$ is *regular* if and only if it is recognized by some DFA. Lewis and Papadimitriou (1981) and Sipser (2005) are among the standard introductory texts on automata and formal languages.

In order to apply our results to regular languages, we begin by defining the instance space $\mathcal{X} = \Sigma^*$ and concept class $\mathcal{C} = \cup_{n\geq 1}\mathrm{DFA}(n)$, where $\mathrm{DFA}(n)$ is the set of all DFAs on n states; both are countable. Once the size functions on $\mathcal{X}$ and $\mathcal{C}$ have been specified, Theorem 1 furnishes an embedding $\phi : \mathcal{X} \to \{0,1\}^{\mathbb{N}}$ that renders all regular languages linearly separable. It is instructive to verify that the construction of the canonical embedding yields the following kernel, which we call a *universal regular kernel*:

$$\begin{aligned} K_{\mathrm{REG}}(x,y) &= \langle \phi(x), \phi(y) \rangle \\ &= \mathbb{1}_{\{x=y\}} + \sum_{n=1}^{\min\{|x|,|y|\}} \sum_{A \in \mathrm{DFA}(n)} \mathbb{1}_{\{x \in L(A)\}} \mathbb{1}_{\{y \in L(A)\}}. \end{aligned}$$

In fact, we obtain a novel characterization of the regular languages:

Theorem 17 *A language $L \subseteq \Sigma^*$ is regular if and only if there exists a finite set of strings, $s_i \in \Sigma^*$ with corresponding weights $\alpha_i \in \mathbb{R}$, $i = 1, \ldots, m$ such that*

$$L = \left\{ x \in \Sigma^* : \sum_{i=1}^{m} \alpha_i K_{\mathrm{REG}}(s_i, x) > 0 \right\}. \tag{26}$$

Proof Theorem 1 provides an embedding $\phi : \Sigma^* \to \{0,1\}^{\mathbb{N}}$ such that for any regular language L there is a $D \in \mathbb{N}$ so that the sets $\phi_D(L)$ and $\phi_D(\Sigma^* \setminus L)$ are separable by a hyperplane in $\{0,1\}^D$, where ϕ_D is the restriction of ϕ to $\{0,1\}^D$. Thus, by Theorem 2, (26) holds for some finite collection of strings s_i and weights α_i. Conversely, suppose that $L \subseteq \Sigma^*$ is expressible as in (26). Writing $w = \sum_{i=1}^m \alpha_i \phi_D(x_i) \in \mathbb{R}^D$, consider the function $f : \Sigma^* \to \mathbb{R}$ defined by

$$f(x) = \sum_{j=1}^{D} w_j [\phi_D(x)]_j$$

and note that $L = \{x : f(x) > 0\}$. Observe that f can only take on finitely many real values $\{r_k : k = 1, \ldots, k_f\}$. Let $L_{r_k} \subseteq \Sigma^*$ be defined by

$$L_{r_k} = f^{-1}(r_k).$$

A subset $I \subseteq \{1, 2, \ldots, N\}$ is said to be *r_k-acceptable* if $\sum_{i \in I} w_i = r_k$. Any such r_k-acceptable set corresponds to a set of strings $L_I \subseteq \Sigma^*$ such that

$$L_I = \left(\bigcap_{i \in I} [\phi_D]_i^{-1}(1) \right) \setminus \left(\bigcup_{i \in \{1,\ldots,N\} \setminus I} [\phi_D]_i^{-1}(1) \right).$$

It remains to show that each $[\phi_D]_i^{-1}(1) \subseteq \Sigma^*$ is a regular language, but this follows immediately from Theorem 1 and our choice of size functions $|\cdot|$ and $\|\cdot\|$. Now each L_{r_k} is the union of finitely many r_k-acceptable L_I's, and L is the union of the L_{r_k} for $r_k > 0$. This means that L is a finite Boolean combination of regular languages and is therefore itself regular. ■

The result above is mainly of theoretical interest. In order to obtain an efficient learning algorithm with generalization bounds, we must stratify K_{REG} into a regularized kernel family $\{K_n\}$ and show that these are efficiently computable (or approximable).

For our purposes, $A \in \text{DFA}(n)$ is a directed multigraph on n labeled vertices (states), with edges labeled by elements in Σ, and some of the vertices designated as accepting. Note that a given regular language has infinitely many representations as various DFAs; we make no attempt to minimize automata or identify isomorphic ones, nor do we exclude degenerate or disconnected ones.

The first order of business is to verify that Assumptions 10 and 11 hold for the present choice of $\mathcal{X}$ and $\mathcal{C}$, with the specified size functions:

Lemma 18 *Let*

$$\mathcal{C}_n = \bigcup_{k=1}^{n} \text{DFA}(k).$$

Then

(i) There is a sampling algorithm with running time $T_{\text{SAM}}(\mathcal{C}, n) = O(|\Sigma|n)$ that outputs random automata $A \in \mathcal{C}_n$, each with probability $|\mathcal{C}_n|^{-1}$

(ii) $|\mathcal{C}_n|$ is computable in time $O(1)$

(iii) for each $x \in \mathcal{X}$ and $A \in \text{DFA}(k)$, the characteristic function $\chi_A(x) = 2\mathbb{1}_{\{x\in L(A)\}} - 1$ is computable in time $O(|x|)$.

Proof Under our definition of $\text{DFA}(k)$, each member is specified by a vertex-labeled directed graph. Equivalently, $A \in \text{DFA}(k)$ is described by the matrix $\delta \in Q^{Q\times\Sigma}$, where $Q = \{1, 2, \ldots, k\}$, and by the set of accepting states $F \subseteq Q$. This immediately implies that

$$|\,\text{DFA}(k)| = 2^k k^{|\Sigma|k}$$

and $|\mathcal{C}_n| = \sum_{k=1}^{n} |\,\text{DFA}(k)|$; thus (ii) is proved. To sample a δ uniformly from $Q^{Q\times\Sigma}$, we may independently draw each $\delta(q,\sigma)$ uniformly at random from Q. Since the latter is achievable in constant time, we have that δ may be sampled in time $O(|\Sigma|k)$. Additionally, for each $q \in Q$ we may flip a fair coin to determine whether $q \in F$; this amounts to a uniform sampling of F. We've established that $A \in \text{DFA}(k)$ may be uniformly sampled in time $O(|\Sigma|k)$.

To sample uniformly from $\mathcal{C}_n$, define first the distribution π on $\{1, \ldots, n\}$ by

$$\pi_k = \frac{|\text{DFA}(k)|}{|\mathcal{C}_n|} = \frac{2^k k^{|\Sigma|k}}{\sum_{i=1}^{n} 2^i i^{|\Sigma|i}}.$$

Sampling a $k \in \{1, \ldots, n\}$ according to π can be done in time $O(n)$ and then A is drawn uniformly from $\text{DFA}(k)$ as described above. The resulting automaton is a uniform draw from $\mathcal{C}_n$, which took $O(|\Sigma|n)$ steps to perform; this proves (i).

To prove (iii) we simply recall that an automaton evaluates a string by reading it from beginning to end, each time evolving from state to state as prescribed by δ. ■

We now state our main learnability result for regular languages. The theorem below is a direct consequence of Theorem 15, where $\theta(m) := \lceil m^{1/2} \rceil$ and $T_{\text{SAM}}(\mathcal{C}, n) = O(n)$, as per Lemma 18:

Theorem 19 *Let $L \subseteq \Sigma^*$ be a regular language and suppose that $S \subset \Sigma^*$ is a sequence of m instances sampled independently from an arbitrary distribution P, and labeled according to L. For any $\delta > 0$, there is a randomized algorithm* `RegLearn`*, which outputs a classifier $\hat{f} : \Sigma^* \to \{-1, 1\}$ such that with probability at least $1 - \delta$ it achieves a small generalization error,*

$$\begin{aligned} P\{\hat{f}(x) \neq \chi_L(x)\} \quad &\leq \quad \frac{2}{m}\left(4R(L)\log(8em)\log(32m) + \log(16m/\delta)\right) \\ &\qquad \textit{for } m > \max\{\text{size}(L)^2, D(L)/2, R(L)^2/8.3 \times 10^4\}, \end{aligned}$$

where $\text{size}(L)$ is the number of states in the smallest automaton recognizing L and $D(L)$, $R(L)$ are constants depending only on L.

Furthermore, `RegLearn` *has deterministic polynomial complexity, with running time*

$$O\left(\ell_{\max} m^{5/2} \log(m/\delta)\right),$$

where $\ell_{\max} := \max\{|s| : s \in S\}$).

8. Empirical Results

In this section we present some preliminary empirical results that provide a proof of concept for our approach. We performed several types of experiments.

8.1 Proof of Concept, Comparisons

In this basic setup, we draw a sample S of $m = 300$ strings in $\{0,1\}^*$ uniformly with mean Poisson length 15 (that is, the string length ℓ is a Poisson random variable with mean 15 and the string x is drawn uniformly from $\{0,1\}^\ell$). The target DFA A is uniformly drawn at random from DFA(n) as described in Lemma 18, where its size is chosen uniformly in the interval $3 \leq n \leq 50$. This DFA is then minimized, so that its number of states represents its "true" complexity. The automaton A is run on the sample S to produce the label vector $Y \in \{-1,1\}^S$.

For the learning process, we randomly sample a fixed number $T = 1000$ of "feature" automata A_i, from the set DFA$(1:18) = \cup_{n=1}^{18}$ DFA(n), where we chose $\theta(m) = \sqrt{m}$ (note that $18 \approx \sqrt{300}$). For each $s \in S$, we compute the embedding vector $\phi(s) \in \{0,1\}^T$ by

$$[\phi(s)]_i = \mathbb{1}_{\{s \in L(A_i)\}}$$

and arrange the $\{\phi(s)\}_{s \in S}$ into the columns of the $T \times m$ matrix Φ. Now that (Φ, Y) corresponds to m labeled points in $\mathbb{R}^T$, we can apply some standard classification algorithms:

- `SVM` finds the maximum-margin hyperplane separating the positive and negative examples
- `AdaBoost` (Freund and Schapire, 1996) uses the feature automata $\{A_i : 1 \leq i \leq T\}$ as weak classifiers and combines them into a weighted sum

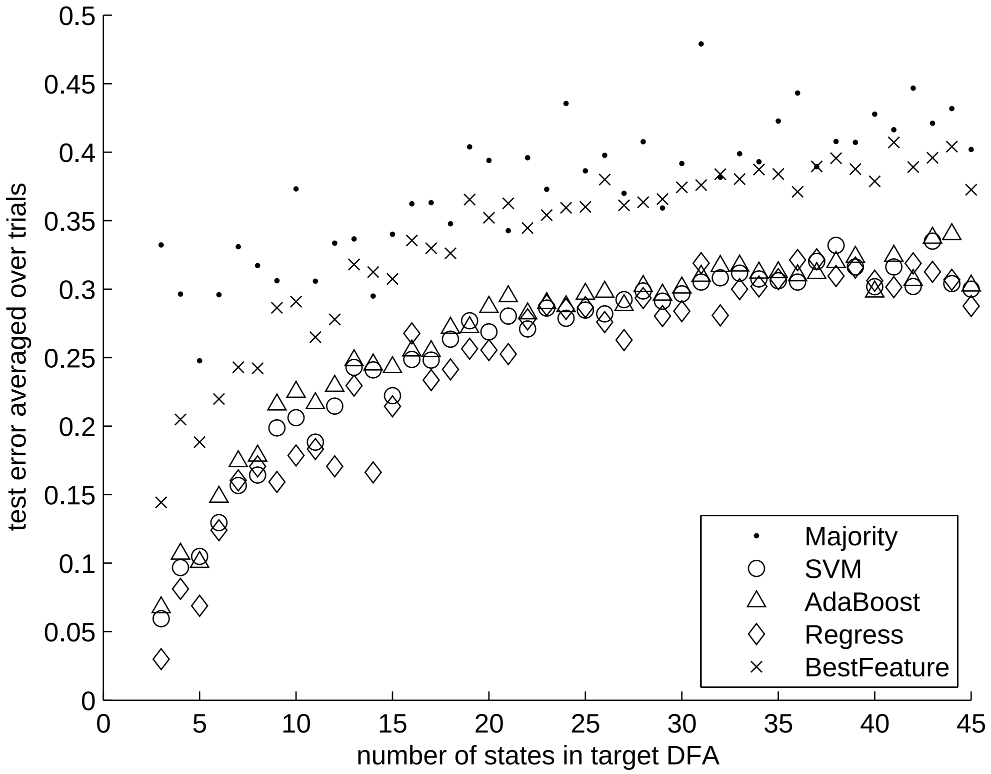

Figure 1: Performance plots of different linear threshold classifiers (sample size = 300, number of features = 1000)

- `Regress` labels a new string $x \in \{0,1\}^*$ as follows:

$$y = \operatorname{sgn}(\phi(x)(\Phi\Phi^{\mathsf{T}})^{-1}\Phi Y)$$

  where $\cdot^{\mathsf{T}}$ denotes the matrix transpose

as well as two "baseline" classifiers:

- `Majority` labels every unseen string $x \in \{0,1\}^*$ by the majority vote of the training labels, disregarding the actual training strings:

$$y = \operatorname{sgn}\left(\sum_{s\in S} Y_s\right)$$

- `BestFeature` picks the single feature automaton A_i with the best empirical performance.

A test set of 100 strings is drawn from the same distribution as S, and the performance of each classifier is compared against the true labels given by A. The results are summarized in Figure 1, averaged over several hundred trials.

As expected, since in this experiment the sample size is kept fixed, the performance degrades with increasing automaton complexity. Furthermore, `SVM`, `AdaBoost` and `Regress` have a similar performance. While the first two methods are margin driven with guaranteed performance bounds, the success of `Regress` is somewhat surprising. We currently have no explanation for why such a seemingly naive approach should work so well. Unsurprisingly, `BestFeature` and `Majority` trail behind the other methods significantly. However, even the latter simplistic approaches perform quite better than chance on moderately sized automata.

Despite the rather pessimistic impossibility results for learning general automata, our results seem to indicate that random automata are efficiently learnable. This is in line with the empirical observations of Lang (1992) and others that random automata are “easy” to learn in practice.

8.2 Role of Adaptive Feature Set

In the second experiment, we demonstrate the importance of having an adaptive feature set. Here, we chose a single fixed target DFA on 20 states, and study the generalization performance of two schemes, one with a fixed number of features T, and one where T is chosen adaptively as a function of sample size. The training and test strings are sampled according to the same scheme as described above. The sample size m varies from 10 to 400; the test set is fixed at 100. In the fixed T scheme we set $T \equiv 500$ while in the adaptive one $T = m \log m$. As expected, the adaptive scheme eventually outperforms the fixed one (see Figure 2).

8.3 Case Study: Parity Languages

For $I = \{i_1 < i_2 \ldots < i_k\} \subset \mathbb{N}$, define the language $L[I] \subset \{0,1\}^*$ by

$$L[I] = \{x \in \{0,1\}^* : |x| \geq i_k, x_{i_1} \oplus x_{i_2} \oplus \ldots \oplus x_{i_k} = 1\}$$

where $\oplus$ is addition modulo 2. Then $L[I]$ is called a *parity language*. The following facts are easily verified

Claim 20 *For $I \subset \mathbb{N}$ and $L[I]$, we have*

(a) the language $L[I]$ is regular

(b) the minimal DFA accepting $L[\{1,k\}]$ has size $2k+1$.

The minimal DFA for the language

$$L[\{1,2\}] = \{x \in \{0,1\}^* : x_1 \oplus x_2 = 1\}$$

is shown in Figure 3. We describe the results of some empirical investigations with parity languages.

8.3.1 EXACT RECOVERY

Recall that the learner is presented with a labeled sample (S,Y), which is consistent with some target language $L_0 \subset \{0,1\}^*$. PAC learning entails producing a hypothesis $\hat{L}$ whose predictions on new strings will be statistically similar to those of L_0. *Exact recovery* is a more stringent demand: we are required to produce a hypothesis that is identical to the target language.

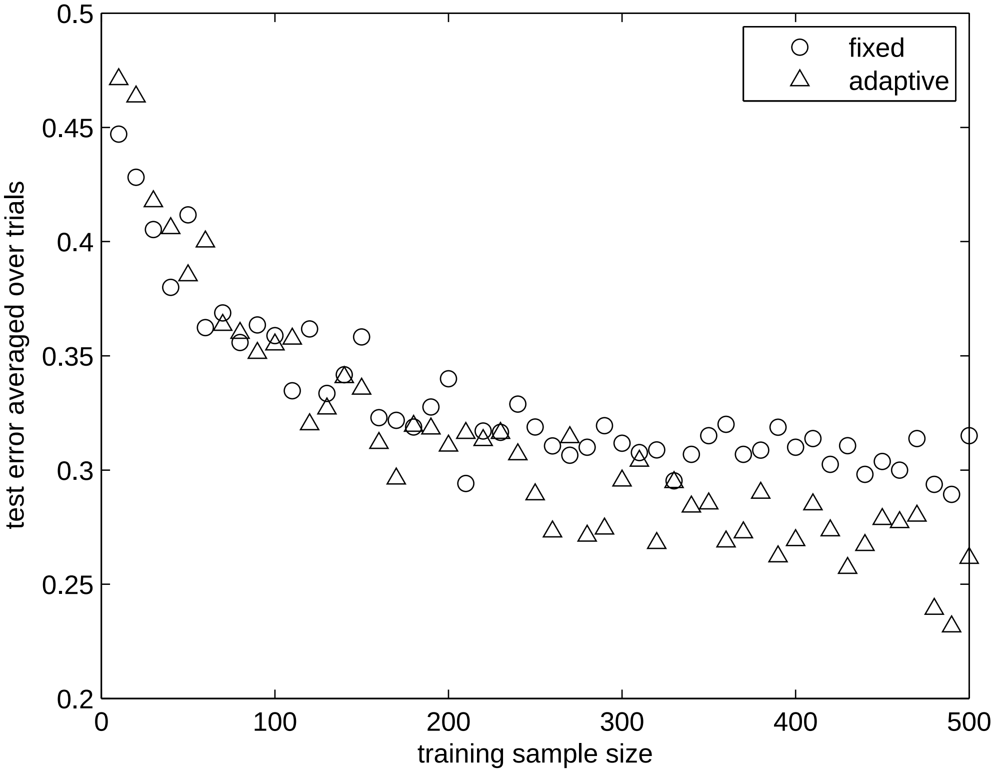

Figure 2: Performance in the fixed and adaptive settings

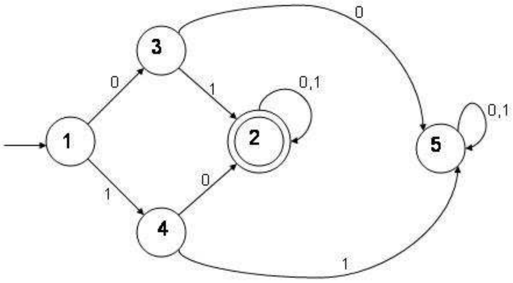

Figure 3: The minimal DFA for the parity language $L[\{1,2\}]$

Note that the output of our algorithm is not an explicit automaton. Instead, we output a collection of DFAs $\{A_1, A_2, \ldots, A_T\}$ with corresponding weights $\alpha_t \in \mathbb{R}$, with the hypothesis language $\hat{L}$ given by

$$\hat{L} = \left\{ x \in \{0,1\}^* : \sum_{t=1}^{T} \alpha_t \mathbb{1}_{\{x \in L(A_t)\}} > 0 \right\}. \tag{27}$$

An interesting question is the relation between $\hat{L}$ and the true unknown language L_0. In principle, the DFA for $\hat{L}$ can be recovered exactly, using the algorithmic construction given in the proof of Theorem 17. However, this procedure has exponential complexity in T and is certainly infeasible for $T \sim 100$. Thus, rather than recovering the *exact* DFA corresponding to $(\{A_t\}, \{\alpha_t\})$, we implement Angluin's algorithm for learning DFAs from membership and equivalence queries (Angluin, 1982). Testing whether a string x belongs to $\hat{L}$ is achievable in time $O(T|x|)$, so membership queries pose no problem. As discussed above, testing whether Angluin's hypothesis language L_{ANGL} is equivalent to $\hat{L}$ is infeasible (at least using the brute-force construction; we do not exclude the possibility of some clever efficient approach). Instead, we only compare L_{ANGL} and $\hat{L}$ on the labeled sample (S, Y). If $\chi_{L_{\text{ANGL}}}(x) = Y_x$ for all $x \in S$, we declare the two equal; otherwise, the first string on which the two disagree is fed as a counterexample into Angluin's algorithm.

We drew a sample of 300 strings in $\{0,1\}^*$ uniformly with mean Poisson length 10, as described in Section 8.1. These strings were labeled consistently with $L[\{1,2\}]$. On this labeled sample we ran the algorithm `RegLearn`, as defined in Theorems 15, and 19, which outputs the language $\hat{L}$, as given in (27). The hypothesis $\hat{L}$ attained an accuracy of 91% on unseen strings and our adaptation of Angluin's algorithm recovered $L[\{1,2\}]$ *exactly* 88% of the time. When this experiment was repeated on $L[\{1,3\}]$, whose minimal DFA has 7 states, prediction accuracy of `RegLearn` dropped to 86% while the target automaton was recovered exactly only 1% of the time. It seems that relatively large sample sizes are needed for exact recovery, though we are not able to quantify (or even conjecture) a rate at this point.

8.3.2 EMPIRICAL MARGIN

We took the "complete" sample $S = \{0,1\}^{\leq 7}$ (i.e., $|S| = 255$) and labeled it with $L[\{1,k\}]$, for $k = 2,3,4,5$. The algorithm `RegLearn` (as a special case of `UnivLearn`) finds the optimal empirical margin $\tilde{\gamma}_*$ as defined in (23). We repeated this experiment a few dozen times—since the kernel is random, the margins obtained are also random. The results are presented in Figure 4, but should be interpreted with caution. As expected, the margin is decreasing with automaton size (the latter given by $2k+1$, as per Claim 20(b)). It is difficult to discern a trend (polynomial/exponential decay) from four points, and extending the experiment for moderate-sized k is computationally expensive.

8.3.3 LONG INPUT STRINGS

One of the claimed advantages of our method is that long training strings do not significantly affect hypothesis complexity. To test this claim empirically, we fixed the target language at $L[\{1,2\}]$ and let the mean string length λ vary from 10 to 100 in increments of 10. For each value of λ, we drew a sample S of 300 strings in $\{0,1\}^*$ uniformly with mean Poisson length λ (as described in Section 8.1) and labeled S consistently with $L[\{1,2\}]$. On this labeled sample we ran the algorithm `RegLearn`, and tested its prediction accuracy on 100 new strings. The results are displayed in Figure 5 and show a graceful degradation of performance. In particular, for strings of mean length

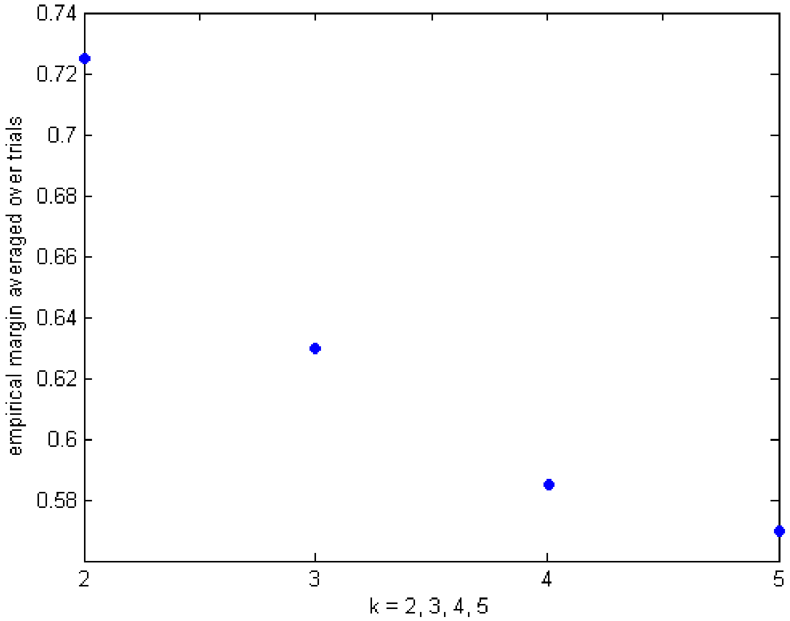

Figure 4: Empirical margins of $L[\{1,k\}]$, for $k = 2,3,4,5$

100, the learner performs significantly better than chance—even though learning parity languages from long strings entails high-dimensional feature selection. This example illustrates the tradeoff between information and computational complexity. If we had unbounded resources, exhaustive enumeration over all automata would most likely find the correct automaton. It is not clear how standard methods, such as the RPNI algorithm (Oncina and Garcia, 1992) would process input strings of such length.

9. Inherent Limitations

An immediate question is, How does the “margin learning rate” $R(\cdot)$ appearing in Theorems 3 and 15 relate to the target concept complexity (say, as measured by $\|\cdot\|$)? One might hope for a bound of the type

$$R(c) \quad = \quad O(\mathrm{poly}(\|c\|)).$$

Unfortunately, under standard cryptographic assumptions this cannot hold in general, as we demonstrate for the case of DFAs. For reference, let us state the Discrete Cube Root (DCR) assumption as it appears in Kearns and Vazirani (1997). In what follows, $N = pq$ is an n-bit number and p,q are randomly chosen primes so that 3 does not divide $(p-1)(q-1)$. The multiplicative group modulo N is denoted by $\mathbb{Z}_N^*$ and x is chosen uniformly random in $\mathbb{Z}_N^*$. DCR states that for every polynomial r, there is no algorithm with running time $r(n)$ that on input N and $y = x^3 \bmod N$ outputs x with probability exceeding $1/r(n)$ (the probability is taken over p, q, x, and the algorithm’s internal randomization).

The following theorem is implicit in chapters 6 and 7 of Kearns and Vazirani (1997), and is a combination of results in Pitt and Warmuth (1990) and Kearns and Valiant (1994):

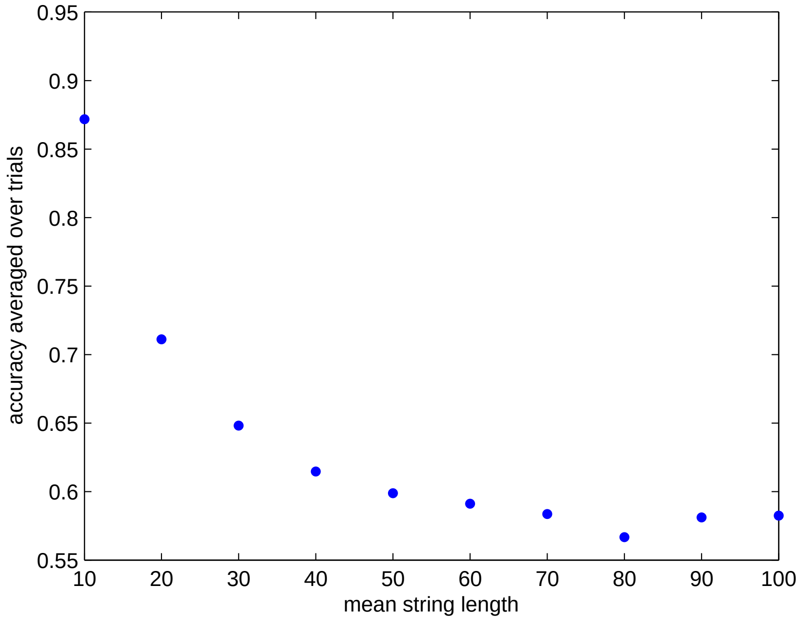

Figure 5: Prediction accuracy when training on long strings

Theorem 21 *For every polynomial p, there is a sequence of distributions P_n on $\{0,1\}^n$ and finite state automata A_n, with $\text{size}(A_n) = \text{poly}(n)$ such that assuming DCR, there is no algorithm with running time $p(n)$ that produces a hypothesis that agrees with A_n on more than $1/2+1/p(n)$ of P_n.*

We will refer to the concept-dependent quantity $R(c)$ appearing in theorems 3, 15, and 19 as the *margin-based rate* and the parameter $D(c)$ appearing in Theorems 15, and 19 as the *intrinsic dimension*. Under the DCR assumption, these quantities cannot both grow polynomially in the target automaton size for *all* automata:

Corollary 22 *If DCR is true, there is no efficiently computable (or approximable) universal regular kernel K and polynomial p such that for every finite state automaton A, we have $\max\{R(A), D(A)\} \leq p(\text{size}(A))$ under K.*

Proof We argue by contradiction. Suppose K is an efficiently computable universal regular kernel, and that for some polynomial p we have the margin-based rate $R(A) \leq p(\text{size}(A))$ under K, for every DFA A. Then by Theorem 3, a sample of size $p(\text{size}(A))^2$ suffices to guarantee (with high probability) the existence of a classifier with polynomially small generalization error, and there is an efficient algorithm (e.g., SVM) to discover such a classifier. This contradicts Theorem 21. A similar contradiction is obtained via Theorem 15 if K is efficiently approximable. ■

Still focusing on the special case of DFAs, another natural line of inquiry is the relationship between the true target language L and the hypothesis language $\hat{L}$ induced by our learning algorithm (note that $\hat{L}$ is necessarily regular, by Theorem 17). In particular, it would be desirable to have a handle on the complexity of the minimal DFA accepting $\hat{L}$ in terms of the corresponding minimal

DFA for L. To be more concrete, suppose we have observed a finite sample $S \subset \Sigma^*$ with labels $Y \in \{-1,1\}^S$. We compute a maximum-margin hyperplane (i.e., a set of weights $\alpha \in \mathbb{R}^S$) for this sample, and consider the resulting language

$$L(S,\alpha) \quad = \quad \{x \in \mathcal{X} : \sum_{s \in S} \alpha_s \tilde{K}(s,x) > 0\}$$

where $\tilde{K}$ is the approximate kernel constructed by the algorithm of Theorem 15; note that $L(S,\alpha)$ is regular. Let $A_0(S,Y)$ be the smallest DFA consistent with the labeled sample (S,Y) and let $A_0(S,\alpha)$ denote the smallest DFA recognizing $L(S,\alpha)$. Now approximately recovering $A_0(S,\alpha)$ is feasible via active learning (see Chapter 8 of Kearns and Vazirani 1997). Given a hyperplane representation for a regular language one may efficiently query the resulting classifier on any string in $x \in \Sigma^*$ to see if $x \in L(S,\alpha)$. Using membership and equivalence queries, Angluin's algorithm (Angluin, 1987) recovers $A_0(S,\alpha)$ exactly. If only membership queries are allowed, we can draw enough strings $x \in \Sigma^*$ according to a distribution P to simulate an equivalence query efficiently, to arbitrary P-accuracy. Note that $L(A)$ and $L(A')$ can differ by only one (very long) word, while the two automata may be of exponentially different size. We do not currently have either (a) an efficient method of exactly recovering $A_0(S,\alpha)$ nor can we claim that (b)

$$\mathrm{size}(A_0(S,\alpha)) \quad \leq \quad \mathrm{poly}(\mathrm{size}(A_0(S,Y)));$$

note that if both (a) and (b) were true, that would imply P=NP via the Pitt and Warmuth (1993) hardness of approximation result.

Another limitation of our approach is the possibility of a "Boolean-independent" concept class $\mathcal{C}$, in which no $c \in \mathcal{C}$ may be expressed as a finite Boolean combination of other concepts. One way to construct such a concept class is by taking $\mathcal{X} = \mathbb{N}$ and $\mathcal{C} = \{c_p : p \text{ is prime}\}$, where

$$x \in c_p \quad \Leftrightarrow \quad x \equiv 0 \ (\mathrm{mod}\ p).$$

That $\mathcal{C}$ is Boolean-independent is seen by taking k distinct primes $\{p_i : 1 \leq i \leq k\}$ and an arbitrary $b \in \{0,1\}^k$. Then the number

$$N_b := \prod_{i=1}^{k} (p_i + b_i)$$

is divisible by p_i iff $b_i = 0$; thus the divisibility of N by $p_1, p_2, \ldots, p_{k-1}$ gives no information regarding its divisibility by p_k. In a Boolean-independent class, no concept may be expressed as a finite linear combination of other concepts (the contrapositive of this claim is established while proving the "if" direction of Theorem 17). Alas, Boolean independence is an inevitable feature of nontrivial concept classes:

Theorem 23 *Any infinite concept class $\mathcal{C}$ contains an infinite Boolean-independent subset.*

Proof Suppose $\mathcal{C}$ contains no infinite Boolean-independent subsets. This means that every $c \in \mathcal{C}$ may be expressed as a finite Boolean combination of elements $\{e_i\}$ from some finite $E \subset \mathcal{C}$. But the Boolean closure of a finite collection of sets is finite, so $\mathcal{C}$ must be finite. ■

The moral of this section is that our approach is certainly vulnerable to the same classical computational hardness reductions as other discrete concept-learning methodologies. One must keep in mind, however, that the hardness reductions are pessimistic, based on carefully contrived target concepts and distributions. As noted in Section 8, "typical" or "random" concepts are often much easier to learn than their worst-case cousins (which is unsurprising given the numerous NP-hard but easy-on-average problems; see, for example Chvátal and Szemerédi 1988 or Flaxman 2003).

10. Discussion

We have presented a new generic technique for learning general countable concept classes and applied it to regular languages. Aside from its generality and ease of implementation, our approach offers certain advantages over existing techniques. Observe that the complexity of the classifier constructed in Theorem 15 does not directly depend on the input string lengths. For the case of learning regular languages, it means that the training string lengths do not affect hypothesis complexity. This is not the case for the methods surveyed in the Introduction, which build a prefix tree acceptor in the initial phase. As already mentioned, we make no structural assumptions on the target automaton (such as acyclic or nearly so), or on the sampling distribution (such as the sample being "structurally complete").

In practice, given *m* samples, our approach attempts to fit the labeled data by a linear combination of concepts with bounded complexity, $\|c\| \leq \theta(m)$. This method may fail for two possible reasons: Either we don't have sufficient number of samples to learn the unknown target concept, or we got unlucky - this specific concept has a very small margin. Note that these sample size restrictions are inherent also in the SRM framework, for example in Theorem 8.

The experiments presented in Section 8 provide support for our approach. Our basic method could be easily adapted to learning context-free and other grammars. There are obvious computability limitations—thus, if our concept class is all Turing-recognizable languages, the corresponding universal kernel is provably uncomputable (Blocki, 2007). Modulo these limitations, any concept class satisfying the efficient sampling and membership evaluation assumptions (Assumptions 10 and 11) is amenable to our approach.

Our work naturally raises some interesting questions—in particular, regarding learning regular languages. We conclude the paper with a few of them.

1. Can K_{REG} be efficiently computed? Is it a #P-complete problem? Is there *any* efficiently computable universal regular kernel?

2. Can $A_0(S,\alpha)$ be efficiently recovered from its hyperplane representation? Can $\text{size}(A_0(S,\alpha))$ be nontrivially bounded in terms of $\text{size}(A_0(S,Y))$? (The notation is from the previous section.)

3. Corollary 22 seems to thwart attempts to bound the margin-based rate for a regular language in terms of the size of its minimal DFA. Is there a different natural complexity measure for regular languages which does allow a polynomial bound on the margin-based rate?

4. The intractability results in Theorem 21 are highly pessimistic, in that both the automaton and the training/testing distribution are contrived to be pathological. What about the complexity of learning "typical" automata—say, those drawn uniformly from $\text{DFA}(n)$? What can be said

about the margin distribution of such automata? Similarly, what about their sample margin, under some natural class of distributions on the strings Σ^*?

Acknowledgments

AK's work at Carnegie Mellon took place under the guidance of John Lafferty, with part of the support from NSF ITR grant IIS-0205456. The research continued at the Weizmann Institute, where AK is being hosted by Gideon Schechtman, and supported in part by the Israel Science Foundation. BN was supported by the Lord Sieff of Brimpton Memorial fund. Various key concepts were crystallized during AK's many valuable discussions with Corinna Cortes and Mehryar Mohri. Thanks also to Jeremiah Blocki, Avrim Blum, Manuel Blum, Daniel Golovin, Uriel Feige, Nati Linial, Alex Smola, Ofer Zeitouni and Noam Zeilberger for helpful and insightful input. We also thank the referees for constructive comments which improved the paper.

References

Dana Angluin. On the complexity of minimum inference of regular sets. *Information and Control*, 3(39):337–350, 1978.

Dana Angluin. Inference of reversible languages. *Journal of the ACM (JACM)*, 3(29):741–765, 1982.

Dana Angluin. Learning regular sets from queries and counterexamples. *Inf. Comput.*, 75(2):87–106, 1987. ISSN 0890-5401. doi: http://dx.doi.org/10.1016/0890-5401(87)90052-6.

Martin Anthony and Peter L. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 1999.

Maria-Florina Balcan, Avrim Blum, and Santosh Vempala. Kernels as Features: On Kernels, Margins, and Low-dimensional Mappings. *Machine Learning*, 65(1):79–94, 2006.

Peter Bartlett, Yoav Freund, Wee Sun Lee, and Robert E. Schapire. Boosting the margin: a new explanation for the effectiveness of voting methods. *Ann. Statist.*, 26(5):1651–1686, 1998.

Jeremiah Blocki. Turing machine kernel is not computable, 2007. Private communication/unpublished note.

Miguel Bugalho and Arlindo L. Oliveira. Inference of regular languages using state merging algorithms with search. *Pattern Recognition*, 38:1457, 2005.

Vašek Chvátal and Endre Szemerédi. Many hard examples for resolution. *J. ACM*, 35(4):759–768, 1988. ISSN 0004-5411. doi: http://doi.acm.org/10.1145/48014.48016.

Alexander Clark and Franck Thollard. Pac-learnability of probabilistic deterministic finite state automata. *Journal of Machine Learning Research (JMLR)*, 5:473–497, 2004.

Corinna Cortes, Leonid Kontorovich, and Mehryar Mohri. Learning languages with rational kernels. *Computational Learning Theory (COLT)*, 2007.

Colin de la Higuera. A bibligraphical study of grammatical inference. *Pattern Recognition*, 38: 1332, 2005.

Abraham Flaxman. A spectral technique for random satisfiable 3cnf formulas. In *SODA '03: Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 357–363, Philadelphia, PA, USA, 2003. Society for Industrial and Applied Mathematics. ISBN 0-89871-538-5.

Yoav Freund and Robert E. Schapire. Predicting $\{0,1\}$-Functions on Randomly Drawn Points. In *Proceedings of the Ninth Annual Conference on Computational Learning Theory (COLT 1996)*, pages 325–332, 1996.

Yoav Freund, Michael Kearns, Dana Ron, Ronitt Rubinfeld, Robert E. Schapire, and Linda Sellie. Efficient learning of typical finite automata from random walks. In *STOC '93: Proceedings of the Twenty-fifth Annual ACM Symposium on Theory of Computing*, pages 315–324, New York, NY, USA, 1993. ACM Press.

Ashutosh Garg, Sariel Har-Peled, and Dan Roth. On generalization bounds, projection profile, and margin distribution. In *ICML '02: Proceedings of the Nineteenth International Conference on Machine Learning*, pages 171–178, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc. ISBN 1-55860-873-7.

E. Mark Gold. Language identification in the limit. *Information and Control*, 50(10):447–474, 1967.

E. Mark Gold. Complexity of automaton identification from given data. *Information and Control*, 3(37):302–420, 1978.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *American Statistical Association Journal*, 58:13–30, 1963.

Yoshiyasu Ishigami and Sei'ichi Tani. Vc-dimensions of finite automata and commutative finite automata with k letters and n states. *Discrete Applied Mathematics*, 74(3):229–240, 1997.

Michael J. Kearns and Leslie G. Valiant. Cryptographic limitations on learning boolean formulae and finite automata. *Journal of the ACM (JACM)*, 41(1):67–95, 1994.

Micheal Kearns and Umesh Vazirani. *An Introduction to Computational Learning Theory*. The MIT Press, 1997.

George Kimeldorf and Grace Wahba. Some results on Tchebycheffian spline functions. *J. Math. Anal. Appl.*, 33:82–95, 1971. ISSN 0022-247x.

Leonid Kontorovich. A universal kernel for learning regular languages. In *The 5th International Workshop on Mining and Learning with Graphs (MLG 2007)*, Florence, Italy, 2007.

Leonid Kontorovich, Corinna Cortes, and Mehryar Mohri. Learning linearly separable languages. In *Proceedings of The 17th International Conference on Algorithmic Learning Theory (ALT 2006)*, volume 4264 of *Lecture Notes in Computer Science*, pages 288–303, Barcelona, Spain, October 2006. Springer, Heidelberg, Germany.

Kevin J. Lang. Random dfa's can be approximately learned from sparse uniform samples. In *Fifth Conference on Computational Learning Theory*, page 45. ACM, 1992.

Harry R. Lewis and Christos H. Papadimitriou. *Elements of the Theory of Computation*. Prentice Hall, 1981.

Arlindo L. Oliveira and Joao P.M. Silva. Efficient algorithms for the inference of minimum size dfas. *Machine Learning*, 44:93, 2001.

José Oncina and Pedro Garcia. Inferring regular languages in polynomial update time. In *Pattern Recognition and Image Analysis*, page 49. World Scientific, 1992.

José Oncina, Pedro García, and Enrique Vidal. Learning subsequential transducers for pattern recognition interpretation tasks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(5):448–458, 1993.

Leonard Pitt and Manfred Warmuth. Prediction-preserving reducibility. *Journal of Computer and System Sciences*, 41(3):430–467, 1990.

Leonard Pitt and Manfred Warmuth. The minimum consistent DFA problem cannot be approximated within any polynomial. *Journal of the Assocation for Computing Machinery*, 40(1):95–142, 1993.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *NIPS 2007*. MIT Press, 2007.

Jorma Rissanen. *Stochastic Complexity in Statistical Inquiry Theory*. World Scientific Publishing Co., Inc., 1989.

Ronald L. Rivest and Robert E. Schapire. Diversity-based inference of finite automata. In *Proc. 28th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 78–87. IEEE Computer Society Press, Los Alamitos, CA, 1987.

Dana Ron, Yoram Singer, and Naftali Tishby. On the learnability and usage of acyclic probabilistic finite automata. *Journal of Computer and System Sciences*, 56(2):133–152, 1998.

Bernhard Schölkopf and Alex Smola. *Learning with Kernels*. MIT Press, 2002.

Shai Shalev-Shwartz, Yoram Singer, and Nathan Srebro. Pegasos: Primal estimated sub-gradient solver for svm. In *ICML '07: Proceedings of the 24th International Conference on Machine learning*, pages 807–814, New York, NY, USA, 2007. ACM. ISBN 9781595937933. doi: 10.1145/1273496.1273598. URL `http://portal.acm.org/citation.cfm?id=1273598`.

John Shawe-Taylor, Peter L. Bartlett, Robert C. Williamson, and Martin Anthony. A framework for structural risk minimisation. In *COLT*, pages 68–76, 1996.

John Shawe-Taylor, Peter L. Bartlett, Robert C. Williamson, and Martin Anthony. Structural risk minimization over data-dependent hierarchies. *IEEE transactions on Information Theory*, 44: 1926–1940, 1998.

Michael Sipser. *Introduction to the Theory of Computation*. Course Technology, 2005.

Boris A. Trakhtenbrot and Janis M. Barzdin. *Finite Automata: Behavior and Synthesis*, volume 1 of *Fundamental Studies in Computer Science*. North-Holland, Amsterdam, 1973.

Leslie G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

Vladimir N. Vapnik. *Estimation of Dependences Based on Empirical Data*. Springer-Verlag, 1982.

Vladimir N. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, 1998.