Learning Linearly Separable Languages

Leonid Kontorovich[1], Corinna Cortes[2], and Mehryar Mohri[3,2]

[1] Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213
[2] Google Research,
1440 Broadway, New York, NY 10018
[3] Courant Institute of Mathematical Sciences,
251 Mercer Street, New York, NY 10012

Abstract. This paper presents a novel paradigm for learning languages that consists of mapping strings to an appropriate high-dimensional feature space and learning a separating hyperplane in that space. It initiates the study of the linear separability of automata and languages by examining the rich class of piecewise-testable languages. It introduces a high-dimensional feature map and proves piecewise-testable languages to be linearly separable in that space. The proof makes use of word combinatorial results relating to subsequences. It also shows that the positive definite kernel associated to this embedding can be computed in quadratic time. It examines the use of support vector machines in combination with this kernel to determine a separating hyperplane and the corresponding learning guarantees. It also proves that all languages linearly separable under a regular finite cover embedding, a generalization of the embedding we used, are regular.

1 Motivation

The problem of learning regular languages, or, equivalently, finite automata, has been extensively studied over the last few decades.

Finding the smallest automaton consistent with a set of accepted and rejected strings was shown to be NP-complete by Angluin [1] and Gold [12]. Pitt and Warmuth [21] further strengthened these results by showing that even an approximation within a polynomial function of the size of the smallest automaton is NP-hard. These results imply the computational intractability of the general problem of passively learning finite automata within many learning models, including the mistake bound model of Haussler et al. [14] or the PAC-learning model of Valiant [16]. This last negative result can also be directly derived from the fact that the VC-dimension of finite automata is infinite.

On the positive side, Trakhtenbrot and Barzdin [24] showed that the smallest finite automaton consistent with the input data can be learned exactly provided that a uniform complete sample is provided, whose size is exponential in that of the automaton. The worst case complexity of their algorithm is exponential but a better average-case complexity can be obtained assuming that the topology and the labeling are selected randomly [24] or even that the topology is selected adversarially [9].

J. Balcazár, P.M. Long, and F. Stephan (Eds.): ALT 2006, LNAI 4264, pp. 288–303, 2006.

The model of identification in the limit of automata was introduced and discussed by Gold [11]. Deterministic finite automata were shown not to be identifiable in the limit from positive examples [11]. But positive results were given for the identification in the limit of the families of k-reversible languages [2] and subsequential transducers [20]. Some restricted classes of probabilistic automata such as acyclic probabilistic automata were also shown by Ron et al. to be efficiently learnable [22].

There is a wide literature dealing with the problem of learning automata and we cannot survey all these results in such a short space. Let us mention however that the algorithms suggested for learning automata are typically based on a state-merging idea. An initial automaton or prefix tree accepting the sample strings 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.

This work departs from this established paradigm in that it does not use the state-merging technique. Instead, it initiates the study of the linear separation of automata or languages by mapping strings to an appropriate high-dimensional feature space and learning a separating hyperplane, starting with the rich class of *piecewise-testable languages*.

Piecewise-testable languages form a non-trivial family of regular languages. They have been extensively studied in formal language theory [18] starting with the work of Imre Simon [23]. A language L is said to be *n-piecewise-testable*, $n \in \mathbb{N}$, if whenever u and v have the same subsequences of length at most n and u is in L, then v is also in L. A language L is said to be *piecewise testable* if it is n-piecewise-testable for some $n \in \mathbb{N}$.

For a fixed n, n-piecewise-testable languages were shown to be identifiable in the limit by García and Ruiz [10]. The class of n-piecewise-testable languages is finite and thus has finite VC-dimension. To the best of our knowledge, there has been no learning result related to the full class of piecewise-testable languages.

This paper introduces an embedding of all strings in a high-dimensional feature space and proves that piecewise-testable languages are finitely linearly separable in that space, that is linearly separable with a finite-dimensional weight vector. The proof is non-trivial and makes use of deep word combinatorial results relating to subsequences. It also shows that the positive definite kernel associated to this embedding can be computed in quadratic time. Thus, the use of support vector machines in combination with this kernel and the corresponding learning guarantees are examined. Since the VC-dimension of the class of piecewise-testable languages is infinite, it is not PAC-learnable and we cannot hope to derive PAC-style bounds for this learning scheme. But, the finite linear separability of piecewise-testable helps us derive weaker bounds based on the concept of the margin.

The linear separability proof is strong in the sense that the dimension of the weight vector associated with the separating hyperplane is finite. This is related to the fact that a *regular finite cover* is used for the separability of piecewise testable languages. This leads us to study the general problem of separability with other finite regular covers. We prove that languages separated with such regular finite covers are necessarily regular.

The paper is organized as follows. Section 2 introduces some preliminary definitions and notations related to strings, automata, and piecewise-testable languages. Section 3 presents the proof of the finite linear separability of piecewise-testable languages using a subsequence feature mapping. The subsequence kernel associated to this feature mapping is shown to be efficiently computable in Section 4. Section 5 uses margin bounds to examine how the support vector machine algorithm combined with the subsequence kernel can be used to learn piecewise-testable languages. Section 6 examines the general problem of separability with regular finite covers and shows that all languages separated using such covers are regular.

2 Preliminaries

In all that follows, Σ represents a finite alphabet. The length of a string $x \in \Sigma^*$ over that alphabet is denoted by $|x|$ and the complement of a subset $L \subseteq \Sigma^*$ by $\overline{L} = \Sigma^* \setminus L$. For any string $x \in \Sigma^*$, we denote by $x[i]$ the ith symbol of x, $i \leq |x|$. More generally, we denote by $x[i:j]$, the substring of contiguous symbols of x starting at $x[i]$ and ending at $x[j]$.

A string x is a *subsequence* of $y \in \Sigma^*$ if x can be derived from y by erasing some of y's characters. We will write $x \sqsubseteq y$ to indicate that x is a subsequence of y. The relation $\sqsubseteq$ defines a partial order over Σ^*. For $x \in \Sigma^n$, the *shuffle ideal* of x is defined as the set of all strings containing x as a subsequence:

$$\text{Ш}(x) = \{u \in \Sigma^* : x \sqsubseteq u\} = \Sigma^* x[1] \Sigma^* \ldots \Sigma^* x[n] \Sigma^*.$$

The definition of piecewise-testable languages was given in the previous section. An equivalent definition is the following: a language is *piecewise-testable* (PT for short) if it is a finite Boolean combination of shuffle ideals [23].

We will often use the *subsequence feature mapping* $\phi : \Sigma^* \to \mathbb{R}^{\mathbb{N}}$ which associates to $x \in \Sigma^*$ a vector $\phi(x) = (y_u)_{u \in \Sigma^*}$ whose non-zero components correspond to the subsequences of x and are all equal to one:[1]

$$y_u = \begin{cases} 1 & \text{if } u \sqsubseteq x, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

3 Linear Separability of Piecewise-Testable Languages

This section shows that any piecewise-testable language is finitely linearly separable for the subsequence feature mapping.

We will show that every piecewise-testable language is given by some *decision list* of shuffle ideals (a rather special kind of Boolean function). This suffices to

[1] Elements $u \in \Sigma^*$ can be used as indices since Σ^* and $\mathbb{N}$ are isomorphic.

prove the finite linear separability of piecewise-testable languages since decision lists are known to be linearly separable Boolean functions [3].

We will say that a string $u \in \Sigma^*$ is *decisive* for a language $L \subseteq \Sigma^*$, if $Ш(u) \subseteq L$ or $Ш(u) \subseteq \overline{L}$. The string u is said to be *positive-decisive* for L when $Ш(u) \subseteq L$ (*negative-decisive* when $Ш(u) \subseteq \overline{L}$). Note that when u is positive-decisive (negative-decisive),

$$x \in Ш(u) \Rightarrow x \in L \quad (\text{resp. } x \in Ш(u) \Rightarrow x \notin L). \tag{2}$$

Lemma 1 (Decisive strings). *Let $L \subseteq \Sigma^*$ be a piecewise-testable language, then there exists a decisive string $u \in \Sigma^*$ for L.*

Proof. We will prove that this property (existence of a decisive string) holds for shuffle ideals and that it is preserved under the Boolean operations (negation, intersection, union). This will imply that it holds for all finite Boolean combinations of shuffle ideals, i.e., for all PT languages.

By definition, a shuffle ideal $Ш(u)$ admits u as a decisive string. It is also clear that if u is decisive for some PT language L, then u is also decisive for $\overline{L}$. Thus, the existence of a decisive string is preserved under negation. For the remainder of the proof, L_1 and L_2 will denote two PT languages over Σ.

If u_1 is positive-decisive for L_1 and u_2 is positive-decisive for L_2, $Ш(u_1) \cap Ш(u_2) \subseteq L = L_1 \cap L_2$. $Ш(u_1) \cap Ш(u_2)$ is not empty since it contains, for example, $u_1 u_2$. For any string $u \in Ш(u_1) \cap Ш(u_2)$, $Ш(u) \subseteq Ш(u_1) \cap Ш(u_2)$, thus any such u is positive-decisive for L. Similarly, when u_1 is negative-decisive for L_1 and u_2 negative-decisive for L_2 any $u \in Ш(u_1) \cup Ш(u_2)$ is negative-decisive for $L = L_1 \cap L_2$. Finally, if u_1 is positive-decisive for L_1 and u_2 negative-decisive for L_2 then any $u \in Ш(u_2)$ is negative-decisive for $L = L_1 \cap L_2 \subseteq L_1$. This shows that the existence of a decisive string is preserved under intersection.

The existence of a decisive string is also preserved under union. If u_1 is positive-decisive for L_1 and u_2 positive-decisive for L_2, then any $u \in Ш(u_1) \cup Ш(u_2)$ is positive-decisive for $L = L_1 \cup L_2$. Similarly, when u_1 is negative-decisive for L_1 and u_2 negative-decisive for L_2, any $u \in Ш(u_1) \cap Ш(u_2) \neq \emptyset$ is negative-decisive for $L = L_1 \cup L_2$. Lastly, if u_1 is positive-decisive for L_1 and u_2 is negative-decisive for L_2 then any $u \in Ш(u_1)$ is positive-decisive for $L = L_1 \cup L_2$. □

We say that u is *minimally decisive* for L if it admits no proper subsequence $v \sqsubseteq u$ that is decisive for L.

Lemma 2 (Finiteness of the set of minimally-decisive strings). *Let $L \subseteq \Sigma^*$ be a PT language and let $D \subseteq \Sigma^*$ be the set of all minimally decisive strings for L, then D is a finite set.*

Proof. Observe that D is a *subsequence-free* subset of Σ^*: no element of D is a proper subsequence of another. Thus, the finiteness of D follows directly from Theorem 1 below. □

The following result, on which Lemma 2 is based, is a non-trivial theorem of word combinatorics which was originally discovered, in different forms, by Higman [15] in 1952 and Haines [13] in 1969. The interested reader could refer to [19, Theorem 2.6] for a modern presentation.

Theorem 1 ([13, 15]). *Let Σ be a finite alphabet and $L \subseteq \Sigma^*$ a language containing no two distinct strings x and y such that $x \sqsubseteq y$. Then L is finite.*

The definitions and the results just presented can be generalized to decisiveness modulo a set V: we will say that a string u is *decisive modulo some* $V \subseteq \Sigma^*$ if $V \cap Ш(u) \subseteq L$ or $V \cap Ш(u) \subseteq \overline{L}$. As before, we will refer to the two cases as *positive-* and *negative-decisiveness modulo* V and similarly define *minimally decisive strings modulo* V. These definitions coincide with ordinary decisiveness when $V = \Sigma^*$.

Lemma 3 (Finiteness of the set of minimally-decisive strings modulo V). *Let $L, V \subseteq \Sigma^*$ be two PT languages and let $D \subseteq \Sigma^*$ be the set of all minimally decisive strings for L modulo V, then D is a non-empty finite set.*

Proof. Lemma 1 on the existence of decisive strings can be generalized straightforwardly to the case of decisiveness modulo a PT language V: if $L, V \subseteq \Sigma^*$ are PT and $V \neq \emptyset$, then there exists $u \in V$ such that u is decisive modulo V for L. Indeed, by Lemma 1, for any language of the form $Ш(s)$ there exists a decisive string $u \in V \cap Ш(s)$. The generalization follows by replacing $Ш(X)$ with $V \cap Ш(X)$ in the proof of Lemma 1.

Similarly, in view of Lemma 2, it is clear that there can only be finitely many minimally decisive strings for L modulo V. □

Theorem 2 (PT decision list). *If $L \subseteq \Sigma^*$ is PT then L is equivalent to some finite decision list Δ over shuffle ideals.*

Proof. Consider the sequence of PT languages $V_1, V_2, \ldots$ defined according to the following process:

- $V_1 = \Sigma^*$.
- When $V_i \neq \emptyset$, V_{i+1} is constructed from V_i in the following way. Let $D_i \subseteq V_i$ be the nonempty and finite set of minimally decisive strings u for L modulo V_i. The strings in D_i are either all positive-decisive modulo V_i or all negative-decisive modulo V_i. Indeed, if $u \in D_i$ is positive-decisive and $v \in D_i$ is negative-decisive then $uv \in Ш(u) \cap Ш(v)$, which generates a contradiction. Define σ_i as $\sigma_i = 1$ when all strings of D_i are positive-decisive, $\sigma_i = 0$ when they are negative-decisive modulo V_i and define V_{i+1} by:

$$V_{i+1} = V_i \setminus Ш(D_i), \tag{3}$$

with $Ш(D_i) = \bigcup_{u \in D_i} Ш(u)$.

We show that this process terminates, that is $V_{N+1} = \emptyset$ for some $N > 0$. Assume the contrary. Then, the process generates an infinite sequence $D_1, D_2, \ldots$. Construct an infinite sequence $X = (x_n)_{n \in \mathbb{N}}$ by selecting a string $x_n \in D_n$ for any $n \in \mathbb{N}$. By construction, $D_{n+1} \subseteq \overline{Ш(D_n)}$ for all $n \in \mathbb{N}$, thus all strings

x_n are necessarily distinct. Define a new sequence $(y_n)_{n\in\mathbb{N}}$ by: $y_1 = x_1$ and $y_{n+1} = x_{\psi(n)}$, where $\psi : \mathbb{N} \to \mathbb{N}$ is defined for all $n \in \mathbb{N}$ by:

$$\psi(n) = \begin{cases} \min\{k \in \mathbb{N} : \{y_1, \ldots, y_n, x_k\} \text{ is subsequence-free}\}, & \text{if such a } k \text{ exists,} \\ \infty & \text{otherwise.} \end{cases} \tag{4}$$

We cannot have $\psi(n) \neq \infty$ for all $n > 0$ since the set $Y = \{y_1, y_2, \ldots\}$ would then be (by construction) subsequence-free and infinite. Thus, $\psi(n) = \infty$ for some $n > 0$. But then any x_k, $k \in \mathbb{N}$, is a subsequence of an element of $\{y_1, \ldots, y_n\}$. Since the set of subsequences of $\{y_1, \ldots, y_n\}$ is finite, this would imply that X is finite and lead to a contradiction.

Thus, there exists an integer $N > 0$ such that $V_{N+1} = \emptyset$ and the process described generates a finite sequence $D = (D_1, \ldots, D_N)$ of nonempty sets as well as a sequence $\sigma = (\sigma_i) \in \{0,1\}^N$. Let Δ be the decision list

$$(Ш(D_1), \sigma_1), \ldots, (Ш(D_N), \sigma_N). \tag{5}$$

Let $\Delta_n : \Sigma^* \to \{0,1\}$, $n = 1, \ldots, N$, be the mapping defined for all $x \in \Sigma^*$ by:

$$\forall x \in \Sigma^*, \quad \Delta_n(x) = \begin{cases} \sigma_n & \text{if } x \in Ш(D_n), \\ \Delta_{n+1}(x) & \text{otherwise,} \end{cases} \tag{6}$$

with $\Delta_{N+1}(x) = \sigma_N$. It is straightforward to verify that Δ_n coincides with the characteristic function of L over $\bigcup_{i=1}^{n} Ш(D_i)$. This follows directly from the definition of decisiveness. In particular, since

$$V_n = \bigcap_{i=1}^{n-1} \overline{Ш(D_i)} \tag{7}$$

and $V_{N+1} = \emptyset$,

$$\bigcup_{i=1}^{N} Ш(D_i) = \Sigma^*, \tag{8}$$

and Δ coincides with the characteristic function of L everywhere. □

Using this result, we show that a PT language is linearly separable with a finite-dimensional weight vector.

Corollary 1. *For any PT language L, there exists a weight vector $w \in \mathbb{R}^{\mathbb{N}}$ with finite support such that $L = \{x : \operatorname{sgn}(\langle w, \phi(x)\rangle) > 0\}$, where ϕ is the subsequence feature mapping.*

Proof. Let L be a PT language. By Theorem 2, there exists a decision list $(Ш(D_1), \sigma_1), \ldots, (Ш(D_N), \sigma_N)$ equivalent to L where each D_n, $n = 1, \ldots, N$, is a finite set. We construct a weight vector $w = (w_u)_{u \in \Sigma^*} \in \mathbb{R}^{\mathbb{N}}$ by starting with $w = 0$ and modifying its coordinates as follows:

$$\forall u \in D_n, \quad w_u = \begin{cases} +(|\displaystyle\sum_{\{v \in \bigcup_{i=n+1}^{N} D_i : w_v < 0\}} w_v| + 1) & \text{if } \sigma_i = 1, \\ -(|\displaystyle\sum_{\{v \in \bigcup_{i=n+1}^{N} D_i : w_v > 0\}} w_v| + 1) & \text{otherwise,} \end{cases} \tag{9}$$

in the order $n = N, N-1, \ldots, 1$. By construction, the decision list is equivalent to $\{x : \mathrm{sgn}(\langle w, \phi(x)\rangle) > 0\}$. Since each D_n, $n = 1, \ldots, N$, is finite, the weight vector w has only a finite number of non-zero coordinates. □

The dimension of the feature space associated to ϕ is infinite, the next section shows that the kernel associated to ϕ can be computed efficiently however.

4 Efficient Kernel Computation

The positive definite symmetric kernel K associated to the subsequence feature mapping ϕ is defined by:

$$\forall x, y \in \Sigma^*, \quad K(x,y) = \langle \phi(x), \phi(y) \rangle = \sum_{u \in \Sigma^*} [\![u \sqsubseteq x]\!]\, [\![u \sqsubseteq y]\!], \tag{10}$$

where $[\![P]\!]$ represents the 0-1 truth value of the predicate P. Thus, $K(x,y)$ counts the number of subsequences common to x and y, without multiplicity.

This subsequence kernel is closely related to but distinct from the one defined by Lodhi et al. [17]. Indeed, the kernel of Lodhi et al. counts the number of occurrences of subsequences common to x and y. Thus, for example $K(abc, acbc) = 8$, since the cardinal of the set of common subsequences of abc and $acbc$, $\{\epsilon, a, b, c, ab, ac, bc, abc\}$, is 8. But, the kernel of Lodhi et al. (without penalty factor) would instead associate the value 9 to the pair $(abc, acbc)$.

A string with n distinct symbols has at least 2^n possible subsequences, so a naive computation of $K(x,y)$ based on the enumeration of the subsequences of x and y is inefficient. We will show however that $K(x,y)$ can be computed in quadratic time, $O(|\Sigma||x||y|)$, using a method suggested by Derryberry [8] which turns out to be somewhat similar to that of Lodhi et al.

For any symbol $a \in \Sigma$ and a string $u \in \Sigma^*$, define $\mathrm{last}_a(u)$ to be 0 if a does not occur in u and the largest index i such that $u[i] = a$ otherwise. For $x, y \in \Sigma^*$, define K' by:

$$\forall x, y \in \Sigma^*, \quad K'(x,y) = \sum_{u \in \Sigma^+} [\![u \sqsubseteq x]\!]\, [\![u \sqsubseteq y]\!]. \tag{11}$$

Thus, $K'(x,y)$ is the number of nonempty subsequences without multiplicity common to x and y. For any $a \in \Sigma$, define K_a by:

$$\forall x, y \in \Sigma^*, \quad K_a(x,y) = \sum_{u \in \Sigma^* a} [\![u \sqsubseteq x]\!]\, [\![u \sqsubseteq y]\!] \tag{12}$$

be the number of such subsequences ending in a. Then, by definition of K',

$$\forall x, y \in \Sigma^*, \quad K'(x,y) = \sum_{a \in \Sigma} K_a(x,y). \tag{13}$$

By definition, if a does not appear in x and or y, then $K_a(x,y) = 0$. Otherwise, let ua be a common subsequence of x and y with $u \neq \emptyset$, then u is a non-empty subsequence of x and y. Thus,

$$K_a(x, y) = \begin{cases} 0 & \text{if } \mathrm{last}_a(x) = 0 \text{ or } \mathrm{last}_a(y) = 0 \\ 1 + K'(x[1 : \mathrm{last}_a(x) - 1], y[1 : \mathrm{last}_a(y) - 1]) & \text{otherwise,} \end{cases} \tag{14}$$

where the addition of 1 in the last equation accounts for the common subsequence $ua = a$ with $u = \epsilon$ which is not computed by K'. The subsequence kernel K, which does count the empty string ϵ as a common subsequence, is given by $K(x, y) = K'(x, y) + 1$. A straightforward recursive algorithm based on Equation 14 can be used to compute K in time $O(|\Sigma'||x||y|)$, where $\Sigma' \subseteq \Sigma$ is the alphabet reduced to the symbols appearing in x and y.

The kernel of Lodhi et al. [17] was shown to be a specific instance of a rational kernel over the $(+, \times)$ semiring [6]. Similarly, it can be shown that the subsequence kernel just examined is related to rational kernels over the $(+, \times)$ semiring.

5 Learning Linearly Separable Languages

This section deals with the problem of learning PT languages. In previous sections, we showed that using the subsequence feature mapping ϕ, or equivalently a subsequence kernel K that can be computed efficiently, PT languages are finitely linearly separable.

These results suggest the use of a linear separation learning technique such as support vector machines (SVM) combined with the subsequence kernel K for learning PT languages [5, 7, 25]. In view of the estimate of the complexity of the subsequence kernel computation presented in the previous section, the complexity of the algorithm for a sample of size m where $x_{\max}$ is the longest string is in $O(\mathrm{QP}(m)) + m^2 |x_{\max}|^2 |\Sigma|)$, where $\mathrm{QP}(m)$ is the cost of solving a quadratic programming problem of size m, which is at most $O(m^3)$.

We will use the standard margin bound to analyze the behavior of that algorithm. Note however that since the VC-dimension of the set of PT languages is infinite, PAC-learning is not possible and we need to resort to a weaker guarantee.

Let $(x_1, y_1), \ldots, (x_m, y_m) \in X \times \{-1, +1\}$ be a sample extracted from a set X ($X = \Sigma^*$ when learning languages). The margin ρ of a hyperplane with weight vector $w \in \mathbb{R}^{\mathbb{N}}$ over this sample is defined by:

$$\rho = \inf_{i=1,\ldots,m} \frac{y_i \langle w, \phi(x_i) \rangle}{\|w\|}.$$

The sample is linearly separated by w iff $\rho > 0$. Note that our definition holds even for infinite-size samples.

The linear separation result shown for the class of PT languages is in fact strong. Indeed, for any weight vector $w \in \mathbb{R}^{\mathbb{N}}$, let $\mathrm{supp}(w) = \{i : w_i \neq 0\}$ denote the support of w, then the following property holds for PT languages.

Definition 1. *Let C be a concept class defined over a set X. We will say that a concept $c \in C$ is* finitely linearly separable*, if there exists a mapping $\phi : X \to \{0, 1\}^{\mathbb{N}}$ and a weight vector $w \in \mathbb{R}^{\mathbb{N}}$ with* finite support*, $|\mathrm{supp}(w)| < \infty$, such that*

$$c = \{x \in X : \langle w, \phi(x) \rangle > 0\}. \tag{15}$$

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

Note that in general a linear separation in an infinite-dimensional space does not guarantee a strictly positive margin ρ. Points in an infinite-dimensional space may be arbitrarily close to the separating hyperplane and their infimum distance could be zero. However, finitely linear separation does guarantee a strictly positive margin.

Proposition 1. *Let C be a class of concepts defined over a set X that is finitely linearly separable using the mapping $\phi : X \to \{0,1\}^{\mathbb{N}}$ and a weight vector $w \in \mathbb{R}^{\mathbb{N}}$. Then, the margin ρ of the hyperplane defined by w is strictly positive, $\rho > 0$.*

Proof. By assumption, the support of w is finite. For any $x \in X$, let $\phi'(x)$ be the projection of $\phi(x)$ on the span of w, $\mathrm{span}(w)$. Thus, $\phi'(x)$ is a finite-dimensional vector for any $x \in X$ with discrete coordinates in $\{0,1\}$. Thus, the set of $S = \{\phi'(x) : x \in X\}$ is finite. Since for any $x \in X$, $\langle w, \phi(x)\rangle = \langle w, \phi'(x)\rangle$, the margin is defined over a finite set:

$$\rho = \inf_{x \in X} \frac{y_x \langle w, \phi'(x)\rangle}{\|w\|} = \min_{z \in S} \frac{y_x \langle w, z\rangle}{\|w\|} > 0, \tag{16}$$

and is thus strictly positive. □

The following general margin bound holds for all classifiers consistent with the training data [4].

Theorem 3 (Margin bound). *Define the class $\mathcal{F}$ of real-valued functions on the ball of radius R in $\mathbb{R}^n$ as*

$$\mathcal{F} = \{x \mapsto \langle w, x\rangle : \|w\| \leq 1, \|x\| \leq R\}. \tag{17}$$

There is a constant α_0 such that, for all distributions D over X, with probability at least $1-\delta$ over m independently generated examples, if a classifier $\mathrm{sgn}(f)$, with $f \in \mathcal{F}$, has margin at least ρ on the training examples, then the generalization error of $\mathrm{sgn}(f)$ is no more than

$$\frac{\alpha_0}{m}\left(\frac{R^2}{\rho^2}\log^2 m + \log(\frac{1}{\delta})\right). \tag{18}$$

Note that the notion of linear separability with a finite sample may be weak. Any sample of size m can be trivially made linearly separable by using an embedding $\phi : X \to \{0,1\}^{\mathbb{N}}$ mapping each point x to a distinct dimension. However, the support of the weight vector increases with the size of the sample and is not bounded. Also, the margin ρ for such a mapping is $\frac{1}{2\sqrt{m}}$ and thus goes to zero as m increases, and the ratio $(R/\rho)^2$, where $R = 1$ is the radius of the sphere containing the sample points, is $(R/\rho)^2 = 4m$. Thus, such trivial linear separations do not guarantee convergence. The bound of Theorem 3 is not effective with that value of $(R/\rho)^2$.

But, the result of the previous sections guarantee linear separability for samples of infinite size with strictly positive margin.

Theorem 4. *Let C be a finitely linearly separable concept class over X with a feature mapping $\phi : X \to \{0,1\}^{\mathbb{N}}$. Define the class $\mathcal{F}$ of real-valued functions on the ball of radius R in $\mathbb{R}^n$ as*

$$\mathcal{F} = \{x \mapsto \langle w, \phi(x)\rangle : \|w\| \leq 1, \|\phi(x)\| \leq R\}. \tag{19}$$

There is a constant α_0 such that, for all distributions D over X, for any concept $c \in C$, there exists $\rho_0 > 0$ such that with probability at least $1 - \delta$ over m independently generated examples according to D, there exists a classifier $\mathrm{sgn}(f)$*, with $f \in \mathcal{F}$, with margin at least ρ_0 on the training examples, and generalization error no more than*

$$\frac{\alpha_0}{m}\left(\frac{R^2}{\rho_0^2}\log^2 m + \log(\frac{1}{\delta})\right). \tag{20}$$

Proof. Fix a concept $c \in C$. By assumption, c is finitely linearly separable by some hyperplane. By Proposition 1, the corresponding margin ρ_0 is strictly positive, $\rho_0 > 0$. ρ_0 is less than or equal to the margin of the optimal hyperplane ρ separating c from $X \setminus c$ based on the m examples.

Since the full sample X is linearly separable, so is any subsample of size m. Let $f \in \mathcal{F}$ be the linear function corresponding to the optimal hyperplane over a sample of size m drawn according to D. Then, the margin of f is at least as large as ρ since not all points of X are used to define f. Thus, the margin of f is greater than or equal to ρ_0 and the statement follows Theorem 3. □

Theorem 4 applies directly to the case of PT languages. Observe that in the statement of the theorem, ρ_0 depends on the particular concept c learned but does not depend on the sample size m.

Note that the linear separating hyperplane with finite-support weight vector is not necessarily an optimal hyperplane. The following proposition shows however that when the mapping ϕ is surjective the optimal hyperplane has the same property.

Proposition 2. *Let $c \in C$ be a finitely linearly separable concept with the feature mapping $\phi : X \to \{0,1\}^{\mathbb{N}}$ and weight vector w with finite support, $|\,\mathrm{supp}(w)| < \infty$, such that $\phi(X) = \mathbb{R}^{\mathbb{N}}$. Assume that ϕ is surjective, then the weight vector $\hat{w}$ corresponding to the optimal hyperplane for c has also a finite support and* $\mathrm{supp}(\hat{w}) \subseteq \mathrm{supp}(w)$.

Proof. Assume that $\hat{w}_i \neq 0$ for some $i \notin \mathrm{supp}(w)$. We first show that this implies the existence of two points $x_- \notin c$ and $x_+ \in c$ such that $\phi(x_-)$ and $\phi(x_+)$ differ only by their ith coordinate.

Let ϕ' be the mapping such that for all $x \in X$, $\phi'(x)$ differs from $\phi(x)$ only by the ith coordinate and let $\hat{w}'$ be the vector derived from $\hat{w}$ by setting the ith coordinate to zero. Since ϕ is surjective, thus $\phi^{-1}(\phi'(x)) \neq \emptyset$. If x and any $x' \in \phi^{-1}(\phi'(x))$ are in the same class for all $x \in X$, then

$$\mathrm{sgn}(\langle \hat{w}, \phi(x)\rangle) = \mathrm{sgn}(\langle \hat{w}, \phi'(x)\rangle). \tag{21}$$

Fix $x \in X$. Assume for example that $[\phi'(x)]_i = 0$ and $[\phi(x)]_i = 1$, then $\langle \hat{w}, \phi'(x) \rangle = \langle \hat{w}', \phi(x) \rangle$. Thus, in view of Equation 21,

$$\mathrm{sgn}(\langle \hat{w}, \phi(x) \rangle) = \mathrm{sgn}(\langle \hat{w}, \phi'(x) \rangle) = \mathrm{sgn}(\langle \hat{w}', \phi(x) \rangle). \tag{22}$$

We obtain similarly that $\mathrm{sgn}(\langle \hat{w}, \phi(x) \rangle) = \mathrm{sgn}(\langle \hat{w}', \phi(x) \rangle)$ when $[\phi'(x)]_i = 1$ and $[\phi(x)]_i = 0$. Thus, for all $x \in X$, $\mathrm{sgn}(\langle \hat{w}, \phi(x) \rangle) = \mathrm{sgn}(\langle \hat{w}', \phi(x) \rangle)$. This leads to a contradiction, since the norm of the weight vector for the optimal hyperplane is the smallest among all weight vectors of separating hyperplanes.

This proves the existence of the $x_- \notin c$ and $x_+ \in c$ with $\phi(x_-)$ and $\phi(x_+)$ differing only by their ith coordinate.

But, since $i \notin \mathrm{supp}(w)$, for two such points $x_- \notin c$ and $x_+ \in c$, $\langle w, \phi(x_-) \rangle = \langle w, \phi(x_+) \rangle$. This contradicts the status of $\mathrm{sgn}(\langle w, \phi(x) \rangle)$ as a linear separator. Thus, our original hypothesis cannot hold: there exists no $i \notin \mathrm{supp}(w)$ such that $\hat{w}_i \neq 0$ and the support of $\hat{w}$ is included in that of w. □

In the following, we will give another analysis of the generalization error of SVMs for finitely separable hyperplanes using the following bound of Vapnik based on the number of essential support vectors:

$$\mathrm{E}[error(h_m)] \leq \frac{\mathrm{E}[(\frac{R_{m+1}}{\rho_{m+1}})^2]}{m+1}, \tag{23}$$

where h_m is the optimal hyperplane hypothesis based on a sample of m points, $error(h_m)$ the generalization error of that hypothesis, R_{m+1} the smallest radius of a set of essential support vectors of an optimal hyperplane defined over a set of $m+1$ points, and ρ_{m+1} its margin.

Let c be a finitely separable concept. When the mapping ϕ is surjective, by Proposition 2, the weight vector $\hat{w}$ of the optimal separating hyperplane for c has finite support and the margin ρ_0 is positive $\rho_0 > 0$. Thus, the smallest radius of a set of essential support vectors for that hyperplane is $R = \sqrt{N(c)}$ where $N(c) = |\,\mathrm{supp}(\hat{w})|$. If R_{m+1} tends to R when m tends to infinity, then for all $\epsilon > 0$, there exists M_ϵ such that for $m > M_\epsilon$, $R^2(m) \leq N(c) + \epsilon$. In view of Equation 23 the expectation of the generalization error of the optimal hyperplane based on a sample of size m is bounded by

$$\mathrm{E}[error(h_m)] \leq \frac{\mathrm{E}[(\frac{R_{m+1}}{\rho_{m+1}})^2]}{m+1} \leq \frac{N(c)+\epsilon}{\rho_0^2(m+1)}. \tag{24}$$

This upper bound varies as $\frac{1}{m}$.

6 Finite Cover with Regular Languages

In previous sections, we introduced a feature mapping ϕ, the subsequence mapping, for which PT languages are finitely linearly separable. The subsequence mapping can be defined in terms of the set of shuffle ideals of all strings,

$U_u = Ш(u)$, $u \in \Sigma^*$. A string x can belong only to a finite number of shuffle ideals U_u, which determine the non-zero coordinates of $\phi(x)$. This leads us to consider other such mappings based on other regular sets U_u and investigate the properties of languages linearly separated for such mappings. The main result of this section is that all such linearly separated languages are regular.

6.1 Definitions

Let $U_n \subseteq \Sigma^*$, $n \in \mathbb{N}$, be a countable family of sets, such any string $x \in \Sigma^*$ lies in at least one and at most finitely many U_n. Thus, for all $x \in \Sigma^*$,

$$1 \leq \sum_n \psi_n(x) < \infty,$$

where ψ_n is the characteristic function of U_n:

$$\psi_n(x) = \begin{cases} 1 & \text{if } x \in U_n \\ 0 & \text{otherwise.} \end{cases}$$

Any such family $(U_n)_{n\in\mathbb{N}}$ is called a *finite cover* of Σ^*. If additionally, each U_n is a regular set and Σ^* is a member of the family, we will say that $(U_n)_{n\in\mathbb{N}}$ is a *regular finite cover* (RFC).

Any finite cover $(U_n)_{n\in\mathbb{N}}$ naturally defines a positive definite symmetric kernel K over Σ^* given by:

$$\forall x, y \in \Sigma^*, \quad K(x, y) = \sum_n \psi_n(x)\psi_n(y).$$

Its finiteness, symmetry, and positive definiteness follow its construction as a dot product. $K(x, y)$ counts the number of common sets U_n that x and y belong to.

We may view $\psi(x)$ as an infinite-dimensional vector in the space $\mathbb{R}^{\mathbb{N}}$, in which case we can write $K(x, y) = \langle \psi(x), \psi(y) \rangle$. We will say that ψ is an *RFC-induced embedding.* Any weight vector $w \in \mathbb{R}^{\mathbb{N}}$ defines a language $L(w)$ given by:

$$L(w) = \{x \in \Sigma^* : \langle w, \psi(x) \rangle > 0\}.$$

Note that since Σ^* is a member of every RFC, $K(x, y) \geq 1$.

6.2 Main Result

The main result of this section is that any finitely linearly separable language under an RFC embedding is regular. The converse is clearly false. For a given RFC, not all regular languages can be defined by some separating hyperplane. A simple counterexample is provided with the RFC $\{\emptyset, U, \Sigma^* \setminus U, \Sigma^*\}$ where U is some regular language. For this RFC, U, its complement, Σ^*, and the empty set are linearly separable but no other regular language is.

Theorem 5. *Let $\psi : \Sigma^* \to \{0, 1\}^{\mathbb{N}}$ be an RFC-induced embedding and let $w \in \mathbb{R}^{\mathbb{N}}$ be a finitely supported weight vector. Then, the language $L(w) = \{x \in \Sigma^* : \langle w, \psi(x) \rangle > 0\}$ is regular.*

Proof. Let $f : \Sigma^* \to \mathbb{R}$ be the function defined by:

$$f(x) = \langle w, \psi(x) \rangle = \sum_{i=1}^{N} w_i \psi_i(x), \tag{25}$$

where the weights $w_i \in \mathbb{R}$ and the integer $N = |\operatorname{supp}(w)|$ are independent of x. Observe that f can only take on finitely many real values $\{r_k : k = 1, \ldots, K\}$. Let $L_{r_k} \subseteq \Sigma^*$ be defined by

$$L_{r_k} = f^{-1}(r_k). \tag{26}$$

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

$$L_I = \left(\bigcap_{i \in I} \psi_i^{-1}(1) \right) \setminus \left(\bigcup_{i \in \{1,\ldots,N\} \setminus I} \psi_i^{-1}(1) \right) = \left(\bigcap_{i \in I} U_i \right) \setminus \left(\bigcup_{i \in \{1,\ldots,N\} \setminus I} U_i \right).$$

Thus, L_I is regular because each U_i is regular by definition of the RFC. 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 positive r_k. □

Theorem 5 provides a representation of regular languages in terms of some subsets of $\mathbb{R}^{\mathbb{N}}$. Although we present a construction for converting this representation to a more familiar one such as a finite automaton, our construction is not necessarily efficient. Indeed, for some r_k there may be exponentially many r_k-acceptable L_Is. This underscores the specific feature of our method. Our objective is to learn regular languages efficiently using some representation, not necessarily automata.

6.3 Representer Theorem

Let $S = \{x_j : j = 1, \ldots, m\} \subseteq \Sigma^*$ be a finite set of strings and $\alpha \in \mathbb{R}^m$. The pair (S, α) defines a language $L(S, \alpha)$ given by:

$$L(S, \alpha) = \{x \in \Sigma^* : \sum_{j=1}^{m} \alpha_j K(x, x_j) > 0\}. \tag{27}$$

Let $w = \sum_{j=1}^{m} \alpha_j \psi(x_j)$. Since each $\psi(x_j)$ has only a finite number of non-zero components, the support of w is finite and by Theorem 5, $L(S, \alpha)$ can be seen to be regular. Conversely, the following result holds.

Theorem 6. *Let $\psi : \Sigma^* \to \{0,1\}^{\mathbb{N}}$ be an RFC-induced embedding and let $w \in \mathbb{R}^{\mathbb{N}}$ be a finitely supported weight vector. Let $L(w)$ be defined by $L(w) = \{x \in \Sigma^* : \langle w, \psi(x) \rangle > 0\}$. Then, there exist (x_j), $j = 1, \ldots, m$, and $\alpha \in \mathbb{R}^m$ such that $L(w) = L(S, \alpha) = \{x \in \Sigma^* : \sum_{j=1}^{m} \alpha_j K(x, x_j) > 0\}$.*

Proof. Without loss of generality, we can assume that no cover set $U_n \neq \Sigma^*$, U_n is fully contained in a finite union of the other cover sets $U_{n'}$, $U_{n'} \neq \Sigma^*$. Otherwise, the corresponding feature component can be omitted for linear separation. Now, for any $U_n \neq \Sigma^*$, let $x_n \in U_n$ be a string that does not belong to any finite union of $U_{n'}$, $U_{n'} \neq \Sigma^*$. For $U_n = \Sigma^*$, choose an arbitrary string $x_n \in \Sigma^*$. Then, by definition of the x_n,

$$\langle w, \psi(x) \rangle = \sum_{j=1}^{m} w_j K(x, x_j). \tag{28}$$

This proves the claim. □

This result shows that any finitely linearly separable language can be inferred from a finite sample.

6.4 Further Characterization

It is natural to ask what property of finitely supported hyperplanes is responsible for their inducing regular languages. In fact, Theorem 5 is readily generalized:

Theorem 7. *Let $f : \Sigma^* \to \mathbb{R}$ be a function such that there exist an integer $N \in \mathbb{N}$ and a function $g : \{0,1\}^N \to \mathbb{R}$ such that*

$$\forall x \in \Sigma^*, \quad f(x) = g(\psi_1(x), \psi_2(x), \ldots, \psi_N(x)), \tag{29}$$

Thus, the value of f depends on a fixed finite number of components of ψ. Then, for any $r \in \mathbb{R}$, the language $L = \{x \in \Sigma^* : f(x) = r\}$ is regular.

Proof. Since f is a function of finitely many binary variables, its range is finite. From here, the proof proceeds exactly as in the proof of Theorem 5, with identical definitions for $\{r_k\}$ and L_{r_k}. □

This leads to the following corollary.

Corollary 2. *Let $f : \Sigma^* \to \mathbb{R}$ be a function satisfying the conditions of Theorem 7. Then, for any $r \in \mathbb{R}$, the languages $L_1 = \{x \in \Sigma^* : f(x) > r\}$ and $L_2 = \{x \in \Sigma^* : f(x) < r\}$ are regular.*

7 Conclusion

We introduced a new framework for learning languages that consists of mapping strings to a high-dimensional feature space and seeking linear separation in that space. We applied this technique to the non-trivial case of PT languages and showed that this class of languages is indeed linearly separable and that the corresponding subsequence kernel can be computed efficiently.

Many other classes of languages could be studied following the same ideas. This could lead to new results related to the problem of learning families of languages or classes of automata.

Acknowledgments

Much of the work by Leonid Kontorovich was done while visiting the Hebrew University, in Jerusalem, Israel, in the summer of 2003. Many thanks to Yoram Singer for providing hosting and guidance at the Hebrew University. Thanks also to Daniel Neill and Martin Zinkevich for helpful discussions. This work was supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. The research at CMU was supported in part by NSF ITR grant IIS-0205456. This publication only reflects the authors' views. Mehryar Mohri's work was partially funded by the New York State Office of Science Technology and Academic Research (NYSTAR).

References

1. Dana Angluin. On the complexity of minimum inference of regular sets. *Information and Control*, 3(39):337–350, 1978.
2. Dana Angluin. Inference of reversible languages. *Journal of the ACM (JACM)*, 3(29):741–765, 1982.
3. Martin Anthony. Threshold Functions, Decision Lists, and the Representation of Boolean Functions. Neurocolt Technical report Series NC-TR-96-028, Royal Holloway, University of London, 1996.
4. Peter Bartlett and John Shawe-Taylor. Generalization performance of support vector machines and other pattern classifiers. In *Advances in kernel methods: support vector learning*, pages 43–54. MIT Press, Cambridge, MA, USA, 1999.
5. Bernhard E. Boser, Isabelle Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop of Computational Learning Theory*, volume 5, pages 144–152, Pittsburg, 1992. ACM.
6. Corinna Cortes, Patrick Haffner, and Mehryar Mohri. Rational Kernels: Theory and Algorithms. *Journal of Machine Learning Research (JMLR)*, 5:1035–1062, 2004.
7. Corinna Cortes and Vladimir N. Vapnik. Support-Vector Networks. *Machine Learning*, 20(3):273–297, 1995.
8. Jonathan Derryberry, 2004. Private communication.
9. 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.
10. Pedro García and José Ruiz. Learning k-testable and k-piecewise testable languages from positive data. *Grammars*, 7:125–140, 2004.
11. E. Mark Gold. Language identification in the limit. *Information and Control*, 50(10):447–474, 1967.
12. E. Mark Gold. Complexity of automaton identification from given data. *Information and Control*, 3(37):302–420, 1978.
13. L. H. Haines. On free monoids partially ordered by embedding. *Journal of Combinatorial Theory*, 6:35–40, 1969.
14. David Haussler, Nick Littlestone, and Manfred K. Warmuth. Predicting $\{0,1\}$-Functions on Randomly Drawn Points. In *Proceedings of the first annual workshop on Computational learning theory (COLT 1988)*, pages 280–296, San Francisco, CA, USA, 1988. Morgan Kaufmann Publishers Inc.

15. George Higman. Ordering by divisibility in abstract algebras. *Proceedings of The London Mathematical Society*, 2:326–336, 1952.
16. Micheal Kearns and Umesh Vazirani. *An Introduction to Computational Learning Theory.* The MIT Press, 1997.
17. Huma Lodhi, John Shawe-Taylor, Nello Cristianini, and Chris Watkins. Text classification using string kernels. In Todd K. Leen, Thomas G. Dietterich, and Volker Tresp, editors, *NIPS 2000*, pages 563–569. MIT Press, 2001.
18. M. Lothaire. *Combinatorics on Words*, volume 17 of *Encyclopedia of Mathematics and Its Applications.* Addison-Wesley, 1983.
19. Alexandru Mateescu and Arto Salomaa. *Handbook of Formal Languages, Volume 1: Word, Language, Grammar*, chapter Formal languages: an Introduction and a Synopsis, pages 1–39. Springer-Verlag New York, Inc., New York, NY, USA, 1997.
20. 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.
21. 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.
22. 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.
23. Imre Simon. Piecewise testable events. In *Automata Theory and Formal Languages*, pages 214–222, 1975.
24. Boris A. Trakhtenbrot and Janis M. Barzdin. *Finite Automata: Behavior and Synthesis*, volume 1 of *Fundamental Studies in Computer Science.* North-Holland, Amsterdam, 1973.
25. Vladimir N. Vapnik. *Statistical Learning Theory.* John Wiley & Sons, 1998.