Maximum Margin Algorithms with Boolean Kernels

Roni Khardon*[1] and Rocco A. Servedio**[2]

[1] Department of Computer Science, Tufts University
Medford, MA 02155, USA
roni@cs.tufts.edu
[2] Department of Computer Science, Columbia University
New York, NY 10027, USA
rocco@cs.columbia.edu

Abstract. Recent work has introduced Boolean kernels with which one can learn over a feature space containing all conjunctions of length up to k (for any $1 \leq k \leq n$) over the original n Boolean features in the input space. This motivates the question of whether maximum margin algorithms such as support vector machines can learn Disjunctive Normal Form expressions in the PAC learning model using this kernel. We study this question, as well as a variant in which structural risk minimization (SRM) is performed where the class hierarchy is taken over the length of conjunctions.

We show that such maximum margin algorithms do not PAC learn $t(n)$-term DNF for any $t(n) = \omega(1)$, even when used with such a SRM scheme. We also consider PAC learning under the uniform distribution and show that if the kernel uses conjunctions of length $\tilde{\omega}(\sqrt{n})$ then the maximum margin hypothesis will fail on the uniform distribution as well. Our results concretely illustrate that margin based algorithms may overfit when learning simple target functions with natural kernels.

1 Introduction

1.1 Background

Maximum margin algorithms, notably Support Vector Machines (SVM) [3], have received considerable attention in recent years (see e.g. [21] for an introduction). In their basic form, SVM learn linear threshold hypotheses and combine two powerful ideas. The first idea is to learn using the linear separator which achieves the *maximum margin* on the training data rather than an arbitrary consistent hypothesis. The second idea is to use an implicit feature expansion by a *kernel function*. The kernel $K : X \times X \to \mathbb{R}$, where X is the original space of examples, computes the inner product in the expanded feature space. Given a kernel K

* This work has been partly supported by NSF Grant IIS-0099446.

** Much of this work was done while supported by an National Science Foundation Mathematical Sciences Postdoctoral Research Fellowship.

which corresponds to some expanded feature space, the SVM hypothesis h is (an implicit representation of) the maximum margin linear threshold hypothesis over this expanded feature space rather than the original feature space. SVM theory implies that if the kernel K is efficiently computable then it is possible to efficiently construct this maximum margin hypothesis h and that h itself is efficiently computable. Several on-line algorithms have also been proposed which iteratively construct large margin hypotheses in the feature space, see e.g. [6].

Another major focus of research in learning theory is the question of whether various classes of Boolean functions can be learned by computationally efficient algorithms. The canonical open question in this area is whether there exist efficient algorithms in Valiant's PAC learning model [23] for learning Boolean formulas in Disjunctive Normal Form, or DNF. This question has been open since the introduction of the PAC model nearly twenty years ago, and has been intensively studied by many researchers (see e.g. [1, 2, 4, 7, 8, 10, 12, 14, 15, 18, 22, 24, 25]).

1.2 Can SVMs learn DNF?

In this paper we analyze the performance of maximum margin algorithms when used with Boolean kernels to learn DNF formulas. Several authors [11, 17, 26, 13] have recently proposed a family of kernel functions $K_k : \{0,1\}^n \times \{0,1\}^n \to \mathbb{N}$, where $1 \leq k \leq n$, such that $K_k(x, y)$ computes the number of (monotone or unrestricted) conjunctions of length (exactly or up to) k which are true in both x and y. This is equivalent to expanding the original feature space of n Boolean features to include all such conjunctions.[1] Since linear threshold elements can represent disjunctions, one can naturally view any DNF formula as a linear threshold function over this expanded feature space. It is thus natural to ask whether the K_k kernel maximum margin learning algorithms are good algorithms for learning DNF.

Additional motivation for studying DNF learnability with the K_k kernels comes from recent progress on the DNF learning problem. The fastest known algorithm for PAC learning DNF is due to Klivans and Servedio [12]; it works by explicitly expanding each example into a feature space of monotone conjunctions and explicitly learning a consistent linear threshold function over this expanded feature space. Since the K_k kernel enables us to do such expansions implicitly in a computationally efficient way, it is natural to investigate whether the K_k-kernel maximum margin algorithm yields a computationally efficient algorithm for PAC learning DNF.

We note that it is easily seen that standard convergence bounds on large margin classifiers do not imply that the K_k kernel maximum margin algorithm is an efficient algorithm for PAC learning DNF. Indeed, the bound given by,

[1] This Boolean kernel is similar to the well known polynomial kernel in that all monomials of length up to k are represented. The main difference is that the polynomial kernel assigns weights to monomials which depend on certain binomial coefficients; thus the weights of different monomials can differ by an exponential factor. In the Boolean kernel all monomials have the same weight.

e.g., Theorem 4.18 of [21] only implies nontrivial generalization error for the K_k kernel algorithm if a sample of size $n^{\Omega(k)}$ is used, and with such a large sample the computational advantage of using the K_k kernel is lost. However, such upper bounds do not imply that the K_k kernel maximum margin algorithm must have poor generalization error if run with a smaller sample. The situation is analogous to that of [19] where the generalization error of the Perceptron and Winnow algorithms were studied. For both Perceptron and Winnow the standard bounds gave only an exponential upper bound on the number of examples required to learn various classes, but a detailed algorithm-specific analysis gave positive PAC learning results for Perceptron and negative PAC results for Winnow for the problems considered. Analogously, in this paper we perform detailed algorithm-specific analyses for the K_k kernel maximum margin algorithms.

1.3 Previous work

Khardon *et al.* constructed a simple Boolean function and an example sequence and showed that this sequence causes the K_n kernel perceptron algorithm (i.e. the Perceptron algorithm run over a feature space of all 2^n monotone conjunctions) to make exponentially many mistakes [11]. The current paper differs in several ways from this earlier work: we study the maximum margin algorithm rather than Perceptron, we consider PAC learning from a random sample rather than online learning, and we analyze the K_k kernels for all $1 \leq k \leq n$.

1.4 Our results

Throughout this paper we study the kernels corresponding to all monotone monomials of length up to k, which we denote by K_k. In addition to maximum margin algorithms we also consider a natural scheme of structural risk minimization (SRM) that can be used with this family of Boolean kernels. In SRM, given a hierarchy of classes $C_1 \subseteq C_2 \subseteq \ldots$, one learns with each class separately and uses a cost function combining the complexity of the class with its observed accuracy to choose the final hypothesis. The cost function typically balances various criteria such as the observed error and the (bound on) generalization error. A natural scheme here is to use SRM over the classes formed by K_k with $k = 1, \ldots, n$.[2]

We prove several negative results which establish strong limitations on the ability of maximum margin algorithms to PAC learn DNF formulas (or other simple Boolean classes) using the monomial kernels. Our first result says essentially that for any $t(n) = \omega(1)$, for all $k = 1, \ldots, n$ the K_k kernel maximum margin algorithm cannot PAC learn $t(n)$-term DNF. More precisely, we prove

Result 1: Let $t(n) = \omega(1)$ and let $\epsilon = \frac{1}{4 \cdot 2^{t(n)}}$. There is a $O(t(n))$-term monotone DNF over $t(n)$ relevant variables, and a distribution $\mathcal{D}$ over $\{0,1\}^n$ such that

[2] This is standard practice in experimental work with the polynomial kernel, where typically small values of k are tried (e.g. 1 to 5) and the best is chosen.

for all $k \in \{1, \ldots, n\}$ the K_k maximum margin hypothesis has error larger than ϵ (with overwhelmingly high probability over the choice of a polynomial size random sample from $\mathcal{D}$).

Note that this result implies that the K_k maximum margin algorithms fail even when combined with SRM *regardless of the cost function*. This is simply because the maximum margin hypothesis has error $> \epsilon$ for all k, and hence the final SRM hypothesis must also have error $> \epsilon$.

While our accuracy bound in the above result is small (it is $o(1)$ since $t(n) = \omega(1)$), a simple variant of the construction used for Result 1 also proves:

Result 2: Let $f(x) = x_1$ be the target function. There is a distribution $\mathcal{D}$ over $\{0,1\}^n$ such that for any $k = \omega(1)$ the K_k maximum margin hypothesis has error at least $\frac{1}{2} - 2^{-n^{\Omega(1)}}$ (with overwhelmingly high probability over the choice of a polynomial size random sample from $\mathcal{D}$).

Thus any attempt to learn using monomials of non-constant size can provably lead to overfitting. Note that for any $k = \Theta(1)$, standard bounds on maximum margin algorithms show that the K_k kernel algorithm can learn $f(x) = x_1$ from a polynomial size sample.

Given these strong negative results for PAC learning under arbitrary distributions, we next consider the problem of PAC learning monotone DNF under the uniform distribution. This is one of the few frameworks in which some positive results have been obtained for learning DNF from random examples only (see e.g. [5, 20]). In this scenario a simple variant of the construction for Result 1 shows that learning must fail if k is too small:

Result 3: Let $t(n) = \omega(1)$ and $\epsilon = \frac{1}{4 \cdot 2^{t(n)}}$. There is a $O(t(n))$-term monotone DNF over $t(n)$ relevant variables such that for all $k < t(n)$ the K_k maximum margin hypothesis has error at least ϵ (with probability 1 over the choice of a random sample from the uniform distribution).

On the other hand, we also show that the K_k algorithm fails under the uniform distribution for large k:

Result 4: Let $f(x) = x_1$ be the target function. For any $k = \tilde{\omega}(\sqrt{n})$, the K_k maximum margin hypothesis will have error $\frac{1}{2} - 2^{-\Omega(n)}$ with probability at least 0.028 over the choice of a polynomial size random sample from the uniform distribution.

Note that there is a substantial gap between the “low” values of k (for which learning is guaranteed to fail) and the “high” values of k (for which we show that learning fails with constant probability). We feel that it is of significant interest to characterize the performance of the K_k maximum margin algorithm under the uniform distribution for these intermediate values of k; a discussion of this point is given in Section 5.

Finally, we note here that some of our results can be adapted to give similar negative results for the standard polynomial kernel.

2 Preliminaries

We consider learning Boolean functions over the Boolean cube $\{0,1\}^n$ so that $f : \{0,1\}^n \to \{0,1\}$. It is convenient to consider instead the range $\{-1,1\}$ with 0 mapped to -1 and 1 mapped to 1. This is easily achieved by the transformation $f'(x) = 1-2f(x)$ and since we deal with linear function representations this can be done without affecting the results. For the rest of the paper we assume this representation.

Our arguments will refer to L_1 and L_2 norms of vectors. We use the notation $|x| = \sum |x_l|$ and $\|x\| = \sqrt{\sum x_l^2}$.

Definition 1. *Let* $h : \mathbb{R}^N \to \{-1,1\}$ *be a linear threshold function* $h(x) = sign(W \cdot x - \theta)$ *for some* $W \in \mathbb{R}^N, \theta \in \mathbb{R}$. *The* margin of h on $\langle z, b\rangle \in \mathbb{R}^N \times \{-1,1\}$ *is*

$$m_h(z,b) = \frac{b(W \cdot z - \theta)}{\|W\|}.$$

Note that $|m_h(z,b)|$ is the Euclidean distance from z to the hyperplane $W \cdot x = \theta$.

Definition 2. *Let* $S = \{\langle x^i, b_i\rangle\}_{i=1,\ldots,m}$ *be a set of labeled examples where each* $x^i \in \mathbb{R}^N$ *and each* $b_i \in \{-1,1\}$. *Let* $h(x) = sign(W \cdot x - \theta)$ *be a linear threshold function. The* margin of h on S *is*

$$m_h(S) = \min_{\langle x,b\rangle \in S} m_h(x,b).$$

The maximum margin classifier for S *is the linear threshold function* $h(x) = sign(W \cdot x - \theta)$ *such that*

$$m_h(S) = \max_{W' \in \mathbb{R}^N, \theta' \in \mathbb{R}} \; \min_{\langle x,b\rangle \in S} \frac{b(W' \cdot x - \theta')}{\|W'\|}. \tag{1}$$

The quantity (1) is called the margin of S *and is denoted* m_S.

Note that $m_S > 0$ iff S is consistent with some linear threshold function. If $m_S > 0$ then the maximum margin classifier for S is unique [21].

Let ϕ be a transformation which maps $\{0,1\}^n$ to $\mathbb{R}^N$ and let $K : \{0,1\}^n \times \{0,1\}^n \to \mathbb{R}$ be the corresponding kernel function $K(x,y) = \phi(x) \cdot \phi(y)$. Given a set of labeled examples $S = \{\langle x^i, b_i\rangle\}_{i=1,\ldots,m}$ where each x^i belongs to $\{0,1\}^n$ we write $\phi(S)$ to denote the set of transformed examples $\{\langle \phi(x^i), b_i\rangle\}_{i=1,\ldots,m}$.

We refer to the following learning algorithm as the *K-maximum margin learner*:

- The algorithm first draws a sample $S = \{\langle x^i, f(x^i)\rangle\}_{i=1,\ldots,m}$ of $m = \text{poly}(n)$ labeled examples from some fixed probability distribution $\mathcal{D}$ over $\{0,1\}^n$; here $f : \{0,1\}^n \to \{-1,1\}$ is the unknown function to be learned.
- The algorithm's hypothesis is $h : \{0,1\}^n \to \{-1,1\}$, $h(x) = \text{sign}(W \cdot \phi(x) - \theta)$ where $\text{sign}(W \cdot x - \theta)$ is the maximum margin classifier for $\phi(S)$. Without loss of generality we assume that W is normalized, that is $\|W\| = 1$. We also assume that S contains both positive and negative examples since otherwise the maximum margin classifier is not defined.

SVM theory tells us that if $K(x,y)$ can be computed in poly(n) time then the K-maximum margin learning algorithm runs in poly(n,m) time and the output hypothesis $h(x)$ can be evaluated in poly(n,m) time [21].

Our goal is to analyze the PAC learning ability of various kernel maximum margin learning algorithms. Recall (see e.g. [9]) that a PAC learning algorithm for a class $\mathcal{C}$ of functions over $\{0,1\}^n$ is an algorithm which runs in time polynomial in n and $\frac{1}{\delta}$, $\frac{1}{\epsilon}$ where δ is a confidence parameter and ϵ is an accuracy parameter. We assume here, as is the case throughout the paper, that each function in $\mathcal{C}$ has a description of size poly(n). Given access to random labelled examples $\langle x, f(x)\rangle$ for any $f \in \mathcal{C}$ and any distribution $\mathcal{D}$ over $\{0,1\}^n$, with probability at least $1-\delta$ a PAC learning algorithm must output an efficiently computable hypothesis h such that $\Pr_{x\in\mathcal{D}}[h(x) \neq f(x)] \leq \epsilon$. If an algorithm only satisfies this criterion for a particular distribution such as the uniform distribution on $\{0,1\}^n$, we say that it is a uniform distribution PAC learning algorithm.

Let $\rho_k(n) = \sum_{i=1}^{i=k}\binom{n}{i}$. Note that the number of nonempty monotone conjunctions (i.e. monomials) of size at most k on n variables is $\rho_k(n)$. For $x \in \{0,1\}^n$ we write $\phi_k(x)$ to denote the $\rho_k(n)$-dimensional vector $(x_T)_{T\subseteq\{1,\ldots,n\},1\leq|T|\leq k}$ where $x_T = \prod_{i\in T} x_i$, i.e. the components of $\phi_k(x)$ are all monotone conjunctions of the desired size. We note that for an example $x \in \{0,1\}^n$, the L_1 norm of the expanded example $\phi_k(x)$ is $|\phi_k(x)| = \rho_k(|x|)$.

For $x, y \in \{0,1\}^n$ we write $x \cdot y$ to denote $\sum_{i=1}^n x_i y_i$, i.e. the number of bits which are 1 in both x and y.

Definition 3. *We write $K_k(x,y)$ to denote $\phi_k(x)\cdot\phi_k(y)$. We refer to K_k as the* k-monomials kernel.

The following theorem shows that the k-monomial kernels are easy to compute:

Theorem 1 ([11]). *For all $1 \leq k \leq n$ we have $K_k(x,y) = \sum_{i=1}^k \binom{x\cdot y}{i}$.*

We will frequently use the following observation which is a direct consequence of the Cauchy-Schwarz inequality:

Observation 1 *If $U \in \mathbb{R}^{N_1}$ with $\|U\| = L$ and $I \subseteq \{1,\ldots,N_1\}$, $|I| = N_2$, then $\sum_{i\in I}|U_i| \leq L\cdot\sqrt{N_2}$.*

As a consequence of Observation 1 we have that if $\rho_k(n) = N_1$ is the number of features in the expanded feature space and $|\phi_k(x)| = \rho_k(|x|) = N_2$, then $U\cdot\phi_k(x) \leq L\cdot\sqrt{N_2}$.

3 Distribution-Free Non-Learnability

We give a DNF and a distribution which are such that the k-monomials kernel fails to learn, for all $1 \leq k \leq n$. The DNF we consider is a read once monotone DNF over $t(n)$ variables where $t(n) = \omega(1)$ and $t(n) = O(\log n)$. In fact our results hold for any $t(n) = \omega(1)$ but for concreteness we use $t(n) = \log n$ as a running example. We have

$$f(x) = (x_1 \cdots x_{4\ell^2}) \vee (x_{4\ell^2+1} \cdots x_{8\ell^2}) \vee \cdots \vee (x_{4\ell^3-4\ell^2+1} \cdots x_{4\ell^3}) \qquad (2)$$

where $4\ell^3 = t(n) = \log n$ so that the number of terms $\ell = \Theta(t(n)^{1/3}) = \Theta((\log n)^{1/3})$. For the rest of this section $f(x)$ will refer to the function defined in Equation (2) and ℓ to its size parameter.

A *polynomial threshold function* is defined by a multivariate polynomial $p(x_1, \ldots, x_n)$ with real coefficients. The output of the polynomial threshold function is 1 if $p(x_1, \ldots, x_n) \geq 0$ and -1 otherwise. The degree of the function is simply the degree of the polynomial p. Note that any hypothesis output by the K_k kernel maximum margin algorithm must be a polynomial threshold function of degree at most k. Minsky and Papert [16] (see also [12]) gave the following lower bound on polynomial threshold function degree for DNF:

Theorem 2. *Any polynomial threshold function for $f(x)$ in Equation (2) must have degree at least ℓ.*

The distribution $\mathcal{D}$ on $\{0,1\}^n$ we consider is the following:

- With probability $\frac{1}{2}$ the distribution outputs 0^n.
- With probability $\frac{1}{2}$ the distribution outputs a string $x \in \{0,1\}^n$ drawn from the following product distribution $\mathcal{D}'$: the first $t(n)$ bits are drawn uniformly, and the last $n - t(n)$ bits are drawn from the product distribution which assigns 1 to each bit with probability $\frac{1}{n^{1/3}}$.

For small values of k the result is representation based and does not depend on the sample drawn:

Lemma 1. *If the maximum margin algorithm uses the kernel K_k for $k < \ell$ when learning $f(x)$ under $\mathcal{D}$ then its hypothesis has error greater than $\epsilon = \frac{1}{4 \cdot 2^{t(n)}} = \frac{1}{4n}$.*

Proof. If hypothesis h has error at most $\epsilon = \frac{1}{4 \cdot 2^{t(n)}}$ under $\mathcal{D}$ then clearly it must have error at most $\frac{1}{2 \cdot 2^{t(n)}}$ under $\mathcal{D}'$. Since we are using the kernel K_k, the hypothesis h is some polynomial threshold function of degree at most k which has error $\tau \leq \frac{1}{2 \cdot 2^{t(n)}}$ under $\mathcal{D}'$. So there must be some setting of the last $n - t(n)$ variables which causes h to have error at most τ under the uniform distribution on the first $t(n)$ bits. Under this setting of variables the hypothesis is a degree-k polynomial threshold function on the first $t(n)$ variables. By Minsky and Papert's theorem, this polynomial threshold function cannot compute the target function exactly, so it must be wrong on at least one setting of the first $t(n)$ variables. But under the uniform distribution, every setting of those variables has probability at least $\frac{1}{2^{t(n)}}$. This contradicts $\tau \leq \frac{1}{2 \cdot 2^{t(n)}}$. □

For larger values of k (in fact for all $k = \omega(1)$) we show that the maximum margin hypothesis will with high probability overfit the sample. The following definition captures typical properties of a sample from distribution $\mathcal{D}$:

Definition 4. *A sample S is a $\mathcal{D}$*-typical sample *if*

- *The sample includes the example 0^n.*
- *Any nonzero example x in the sample has $0.99n^{2/3} \leq |x| \leq 1.01n^{2/3}$.*
- *Every pair of positive and negative examples x^i, x^j in S satisfies $x^i \cdot x^j \leq 1.01n^{1/3}$.*

We are interested in cases where a polynomial size sample is used by the algorithm. The following two lemmas hold by standard Chernoff bound arguments:

Lemma 2. *For $m = poly(n)$, with probability $1-2^{-n^{\Omega(1)}}$ a random i.i.d. sample of m draws from $\mathcal{D}$ is a $\mathcal{D}$-typical sample.*

Definition 5. *Let S be a sample. The set $Z(S)$ includes all positive examples z such that every example x in S satisfies $x \cdot z \leq 1.01n^{1/3}$.*

Lemma 3. *Let S be a $\mathcal{D}$-typical sample of size $m = poly(n)$ examples. Then* $\Pr_{\mathcal{D}}[z \in Z(S) | f(z) = 1] = 1 - 2^{-n^{\Omega(1)}}$.

We now show that for a $\mathcal{D}$-typical sample one can achieve a very large margin:

Lemma 4. *Let S be a $\mathcal{D}$-typical sample. Then the maximum margin m_S satisfies*

$$m_S \geq M_{h'} \equiv \frac{1}{2} \cdot \frac{\rho_k(.99n^{2/3}) - m\rho_k(1.01n^{1/3})}{\sqrt{m\rho_k(1.01n^{2/3})}}$$

Proof. We exhibit an explicit linear threshold function h' which has margin at least $M_{h'}$ on the data set. Let $h'(x) = \text{sign}(W' \cdot \phi(x) - \theta')$ be defined as follows:

- $W'_T = 1$ if T is active in some positive example;
- $W'_T = 0$ if T is not active in any positive example.
- θ' is the value that gives the maximum margin on $\phi_k(S)$ for this W', i.e. θ' is the average of the smallest value of $W' \cdot \phi_k(x^{i,+})$ and the largest value of $W' \cdot \phi_k(x^{j,-})$.

Since each positive example x^+ in S has at least $.99n^{2/3}$ ones, we have $W' \cdot \phi(x^+) \geq \rho_k(.99n^{2/3})$. Since each positive example has at most $1.01n^{2/3}$ ones, each positive example in the sample contributes at most $\rho_k(1.01n^{2/3})$ ones to W', so $\|W'\| \leq \sqrt{m\rho_k(1.01n^{2/3})}$. Finally, since each negative example x^- in the sample and each positive example x^+ in the sample share at most $1.01n^{1/3}$ ones, for any x^- in the sample $W' \cdot \phi(x^-) \leq m\rho_k(1.01n^{1/3})$. Putting these conditions together, we get that the margin of h' on the sample is at least

$$\frac{1}{2} \cdot \frac{\rho_k(.99n^{2/3}) - m\rho_k(1.01n^{1/3})}{\sqrt{m\rho_k(1.01n^{2/3})}}$$

as desired. □

Lemma 5. *If S is a $\mathcal{D}$-typical sample, then the threshold θ in the maximum margin classifier for S is at least $M_{h'}$.*

Proof. Let $h(x) = \text{sign}(W \cdot \phi(x) - \theta)$ be the maximum margin hypothesis. Since $\|W\| = 1$ we have

$$\theta = \frac{\theta}{\|W\|} = m_h(\phi_k(0^n), -1) \geq m_{h'}(S) \geq M_{h'}$$

where the second equality holds because $W \cdot \phi(0^n) = 0$ and the last inequality is by Lemma 4. □

Lemma 6. *If the maximum margin algorithm uses the kernel K_k for $k = \omega(1)$ when learning $f(x)$ under $\mathcal{D}$ then with probability $1 - 2^{-n^{\Omega(1)}}$ its hypothesis has error greater than $\epsilon = \frac{1}{4 \cdot 2^{t(n)}} = \frac{1}{4n}$.*

Proof. Let S be the sample used for learning and let $h(x) = \text{sign}(W \cdot \phi_k(x) - \theta)$ be the maximum margin hypothesis. It is well known (see e.g. Proposition 6.5 of [21]) that the maximum margin weight vector W is a linear combination of the support vectors, i.e. of certain examples $\phi_k(x)$ in the sample $\phi_k(S)$. Hence the only coordinates W_T of W that can be nonzero are those corresponding to features (conjunctions) T such that $x_T = 1$ for some example x in S.

By Lemma 2 we have that with probability $1 - 2^{-n^{\Omega(1)}}$ the sample S is $\mathcal{D}$-typical. Consider any $z \in Z(S)$. It follows from the above observations on W that $W \cdot \phi_k(z)$ is a sum of at most $m\rho_k(1.01n^{1/3})$ nonzero numbers, and moreover the sum of the squares of these numbers is at most 1. Thus by Observation 1 we have that $W \cdot \phi_k(z) \leq \sqrt{m\rho_k(1.01n^{1/3})}$. The positive example z is erroneously classified as negative by h if $\theta > W \cdot \phi_k(z)$; by Lemma 5 this inequality holds if

$$\frac{1}{2} \cdot \frac{\rho_k(.99n^{2/3}) - m\rho_k(1.01n^{1/3})}{\sqrt{m\rho_k(1.01n^{2/3})}} > \sqrt{m\rho_k(1.01n^{1/3})},$$

i.e. if

$$\rho_k(.99n^{2/3}) > 2m\sqrt{\rho_k(1.01n^{1/3})\rho_k(1.01n^{2/3})} + m\rho_k(1.01n^{1/3}). \qquad (3)$$

One can show that this equation holds for any $k = \omega(1)$; the proof is omitted for lack of space and will be given in the full version of the paper.

Finally, observe that positive examples have probability at least $\frac{1}{2^{t(n)}} = \frac{1}{n}$. The above argument shows that any $z \in Z(S)$ is misclassified, and Lemma 3 guarantees that the relative weight of $Z(S)$ in positive examples is $1 - 2^{-n^{\Omega(1)}}$. Thus the overall error rate of h under $\mathcal{D}$ is at least $\frac{1}{4 \cdot 2^{t(n)}} = \frac{1}{4n}$ as claimed. □

Together, Lemma 1 and Lemma 6 imply Result 1:

Theorem 3. *For any value of k, if the maximum margin algorithm uses the kernel K_k when learning $f(x)$ under $\mathcal{D}$ then with probability $1 - 2^{-n^{\Omega(1)}}$ its hypothesis has error greater than $\epsilon = \frac{1}{4 \cdot 2^{t(n)}} = \frac{1}{4n}$.*

With a small modification we can also obtain Result 2. In particular, since we do not need to deal with small k we can use a simple function $f = x_1$ and modify $\mathcal{D}$ slightly so that the probability that $f(x) = 1$ is 0.5. Now the argument of Lemma 6 yields

Theorem 4. *For $k = \omega(1)$, if the maximum margin algorithm uses the kernel K_k when learning $f(x) = x_1$ under $\mathcal{D}$ then with probability $1 - 2^{-n^{\Omega(1)}}$ its hypothesis has error at least $\epsilon = \frac{1}{2} - 2^{-n^{\Omega(1)}}$.*

4 Uniform Distribution

While Theorem 3 tells us that the K_k-maximum margin learner is not a PAC learning algorithm for monotone DNF in the distribution-free PAC model, it does not rule out the possibility that the K_k-maximum margin learner might succeed for particular probability distributions such as the uniform distribution on $\{0,1\}^n$. In this section we investigate the uniform distribution.

In Section 3 we took advantage of the fact that 0^n occurred with high weight under the distribution $\mathcal{D}$. This let us give a lower bound (of 0) on the value of $W \cdot \phi_k(x)$ for some negative example in the sample, and we then could argue that the value of θ in the maximum margin classifier must be at least as large as m_S. For the uniform distribution, though, this lower bound no longer holds, so we must use a more subtle analysis.

Before turning to the main result, it is easy to observe that the proof of Lemma 1 goes through for the uniform distribution as well (we actually gain a factor of 2). This therefore proves Result 3: if the algorithm uses too low a degree k then its hypothesis cannot possibly be a sufficiently accurate approximation of the target. In contrast, the next result will show that if a rather large k is used then the algorithm is likely to overfit.

For the next result, we consider the target function $f(x) = x_1$. Let $S = S^+ \cup S^-$ be a data set drawn from the uniform distribution $\mathcal{U}$ and labelled according to the function $f(x)$ where $S^+ = \{\langle x^{i,+}, 1\rangle\}_{i=1,\ldots,m_+}$ are the positive examples and $S^- = \{\langle x^{j,-}, -1\rangle\}_{j=1,\ldots,m_-}$ are the negative examples. Let u_i denote $|x^{i,+}|$ the weight of the i-th positive example, and let the positive examples be ordered so that $u_1 \leq u_2 \leq \cdots \leq u_{m^+}$. Similarly let v_j denote $|x^{j,-}|$ the weight of the j-th negative example with $v_1 \leq v_2 \leq \cdots \leq v_{m^-}$.

Definition 6. *A sample S is a $\mathcal{U}$-*typical sample *if*

- *Every example $x \in S$ satisfies $0.49n \leq |x| \leq 0.51n$.*
- *Every pair of positive and negative examples $x^{i,+}, x^{j,-}$ in S satisfy $x^{i,+} \cdot x^{j,-} \leq 0.26n$.*

A straightforward application of Chernoff bounds yields the next two lemmas:

Lemma 7. *For $m = poly(n)$, with probability $1 - 2^{-\Omega(n)}$ a random i.i.d. sample of m draws from $\mathcal{U}$ is a $\mathcal{U}$-typical sample.*

Definition 7. *Let S be a sample. The set $Z(S)$ includes all positive examples z such that every example x in S satisfies $x \cdot z \leq 0.26n$.*

Lemma 8. *Let S be a $\mathcal{U}$-typical sample of size $m = poly(n)$ examples. Then $\Pr_{\mathcal{U}}[z \in Z(S) | f(z) = 1] = 1 - 2^{-\Omega(n)}$.*

The following lemma is analogous to Lemma 4:

Lemma 9. *Let S be a $\mathcal{U}$-typical sample of size m. Then the maximum margin m_S satisfies*

$$m_S \geq \frac{1}{2}\left(\frac{1}{\sqrt{m}}\sqrt{\rho_k(u_1)} - \sqrt{m\rho_k(.26n)}\right).$$

Proof. We exhibit an explicit linear threshold function h' which has this margin. Let $h'(x) = \mathrm{sign}(W' \cdot \phi_k(x) - \theta')$ be defined as follows:

- For each positive example $x^{i,+}$ in S, pick a set of $\rho_k(u_1)$ features (monomials) which take value 1 on $x^{i,+}$. This can be done since each positive example $x^{i,+}$ has at least u_1 bits which are 1. For each feature T in each of these sets, assign $W'_T = 1$.
- For all remaining features T set $W'_T = 0$.
- Set θ' to be the value that gives the maximum margin on $\phi_k(S)$ for this W', i.e. θ' is the average of the smallest value of $W' \cdot \phi_k(x^{i,+})$ and the largest value of $W' \cdot \phi_k(x^{j,-})$.

Note that since each positive example contributes at most $\rho_k(u_1)$ nonzero coefficients to W', the number of 1's in W' is at most $m\rho_k(u_1)$, and hence $\|W'\| \leq \sqrt{m\rho_k(u_1)}$. By construction we also have that each positive example $x^{i,+}$ satisfies $W' \cdot \phi_k(x^{i,+}) \geq \rho_k(u_1)$.

Since S is a $\mathcal{U}$-typical sample, each negative example $x^{j,-}$ in S shares at most $.26n$ ones with any positive example in S. Hence the value of $W' \cdot \phi_k(x^{j,-})$ is a sum of at most $m\rho_k(.26n)$ numbers whose squares sum to at most $m\rho_k(u_1)$. By Observation 1 we have that $W' \cdot \phi_k(x^{j,-}) \leq \sqrt{m\rho_k(.26n)}\sqrt{m\rho_k(u_1)}$.

The lemma follows by combining the above bounds on $\|W'\|$, $W' \cdot \phi_k(x^{i,+})$ and $W' \cdot \phi_k(x^{j,-})$. □

It turns out that the relative sizes of u_1 and v_1 (the weights of the lightest positive and negative examples in S) play an important role.

Definition 8. *A sample S of size m is* positive-skewed *if $u_1 \geq v_1 + B$, i.e. the lightest positive example in S weighs at least B more than the lightest negative example, where $B = \frac{1}{66}\sqrt{\frac{n}{\log m}}$.*

The following lemma shows that a random sample is positive skewed with constant probability (the proof is omitted for lack of space and is given in the full version of the paper):

Lemma 10. *Let S be a sample of size $m = poly(n)$ drawn from the uniform distribution. Then S is positive-skewed with probability at least* 0.029.

Now we can give a lower bound on the threshold θ for the maximum margin classifier.

Lemma 11. *Let S be a labeled sample of size m which is $\mathcal{U}$-typical and positive skewed, and let $h(x) = sign(W \cdot \phi_k(x) - \theta)$ be the maximum margin hypothesis for S. Then*

$$\theta \geq \frac{1}{2}\left(\frac{1}{\sqrt{m}}\sqrt{\rho_k(u_1)} - \sqrt{m\rho_k(.26n)}\right) - \sqrt{\rho_k(u_1 - B)}.$$

Proof. Since S is positive-skewed we know that $W \cdot \phi_k(x^{1,-})$ is a sum of at most $\rho_k(u_1 - B)$ weights W_T, and since W is normalized the sum of the squares of these weights is at most 1. By Observation 1 we thus have $W \cdot \phi_k(x^{1,-}) \geq -\sqrt{\rho_k(u_1 - B)}$. Since $\theta \geq W \cdot \phi_k(x^{1,-}) + m_S$, together with Lemma 9 this proves the lemma. □

Putting all of the pieces together, we have:

Theorem 5. *If the maximum margin algorithm uses the kernel K_k for $k = \omega(\sqrt{n}\log^{\frac{3}{2}} n)$ when learning $f(x) = x_1$ under the uniform distribution then with probability at least 0.028 its hypothesis has error $\epsilon = \frac{1}{2} - 2^{-\Omega(n)}$.*

Proof. By Lemmas 7 and 10, the sample S used for learning is both $\mathcal{U}$-typical and positive skewed with probability at least $0.029 - 1/2^{-\Omega(n)} > 0.028$. Consider any $z \in Z(S)$. Using the reasoning from Lemma 6, $W \cdot \phi(z)$ is a sum of at most $m\rho_k(.26n)$ numbers whose squares sum to at most 1, so $W \cdot \phi(z) \leq \sqrt{m\rho_k(.26n)}$. The example z is erroneously classified as negative by h if

$$\frac{1}{2}\left(\frac{1}{\sqrt{m}}\sqrt{\rho_k(u_1)} - \sqrt{m\rho_k(.26n)}\right) - \sqrt{\rho_k(u_1 - B)} > \sqrt{m\rho_k(.26n)}.$$

so it suffices to show that

$$\sqrt{\rho_k(u_1)} > 3m\left(\sqrt{\rho_k(.26n)} + \sqrt{\rho_k(u_1 - B)}\right). \tag{4}$$

In Appendix A we show that this holds for all $k = \omega(\sqrt{n}\log^{\frac{3}{2}n})$ as required.

The above argument shows that any $z \in Z(S)$ is misclassified, and Lemma 8 guarantees that the relative weight of $Z(S)$ in positive examples is $1 - 2^{-\Omega(n)}$. Since $\Pr_{x \in \mathcal{U}}[f(x) = 1]$ is $1/2$, we have that with probability at least 0.028 the hypothesis h has error rate at least $\epsilon = \frac{1}{2} - 2^{-\Omega(n)}$, and we are done. □

5 Conclusions and Future Work

Boolean kernels offer an interesting new algorithmic approach to one of the major open problems in computational learning theory, namely learnability of DNF expressions. We have studied the performance of a maximum margin algorithm with the Boolean kernels, giving negative results for several settings of the problem. Our results indicate that the maximum margin algorithm can overfit even when learning simple target functions and using natural and expressive kernels for such functions, and even when combined with structural risk minimization.

We hope that these negative results will be used as a tool to explore alternate approaches which may succeed; we now discuss these briefly.

One direction for future work is to modify the basic learning algorithm. Many interesting variants of the basic maximum margin algorithm have been used in recent years, such as soft margin criteria, kernel regularization, etc.. It may be possible to prove positive results for some DNF learning problems using these approaches. A starting point would be to test their performance on the counterexamples (functions and distributions) which we have constructed.

A more immediate goal is to close the gap between small and large k in our results for the uniform distribution. It is well known [24] that when learning polynomial size DNF under uniform, conjunctions of length $\omega(\log n)$ can be ignored with little effect. Hence the most interesting setting of k for the uniform distribution learning problem is $k = \Theta(\log n)$. Learning under uniform with a $k = \Theta(\log n)$ kernel is qualitatively quite different from learning with the large values of k which we were able to analyze. For example, for $k = \Theta(\log n)$ if a sufficiently large polynomial size sample is taken, then with very high probability all features (monomials of size at most k) are active in the sample.

As a first concrete problem in this scenario, one might consider the question of whether a $k = \Theta(\log n)$ kernel maximum margin algorithm can efficiently PAC learn the target function $f(x) = x_1$ under uniform. For this problem it is easy to show that that the naive hypothesis h' constructed in our proofs achieves both a large margin and high accuracy. Moreover, it is possible to show that with high probability the maximum margin hypothesis has a margin which is within a multiplicative factor of $(1 + o(1))$ of the margin achieved by h'. Though these preliminary results do not answer the above question they suggest that the answer may be positive. A positive answer, in our view, would be strong motivation to analyze the general case.

References

[1] A. Blum, M. Furst, J. Jackson, M. Kearns, Y. Mansour, and S. Rudich. Weakly learning DNF and characterizing statistical query learning using Fourier analysis. In *Proceedings of the Twenty-Sixth Annual Symposium on Theory of Computing*, pages 253–262, 1994.

[2] A. Blum and S. Rudich. Fast learning of k-term DNF formulas with queries. *Journal of Computer and System Sciences*, 51(3):367–373, 1995.

[3] B. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pages 144–152, 1992.

[4] N. Bshouty. A subexponential exact learning algorithm for DNF using equivalence queries. *Information Processing Letters*, 59:37–39, 1996.

[5] N. Bshouty and C. Tamon. On the Fourier spectrum of monotone functions. *Journal of the ACM*, 43(4):747–770, 1996.

[6] C. Gentile. A new approximate maximal margin classification algorithm. *Journal of Machine Learning Research*, 2:213–242, 2001.

[7] T. Hancock and Y. Mansour. Learning monotone k-μ DNF formulas on product distributions. In *Proceedings of the Fourth Annual Conference on Computational Learning Theory*, pages 179–193, 1991.

[8] J. Jackson. An efficient membership-query algorithm for learning DNF with respect to the uniform distribution. *Journal of Computer and System Sciences*, 55:414–440, 1997.

[9] M. Kearns and U. Vazirani. *An introduction to computational learning theory.* MIT Press, Cambridge, MA, 1994.

[10] R. Khardon. On using the Fourier transform to learn disjoint DNF. *Information Processing Letters*, 49:219–222, 1994.

[11] R. Khardon, D. Roth, and R. Servedio. Efficiency versus convergence of boolean kernels for on-line learning algorithms. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, Cambridge, MA, 2002. MIT Press.

[12] A. Klivans and R. Servedio. Learning DNF in time $2^{\tilde{O}(n^{1/3})}$. In *Proceedings of the Thirty-Third Annual Symposium on Theory of Computing*, pages 258–265, 2001.

[13] A. Kowalczyk, A. J. Smola, and R. C. Williamson. Kernel machines and boolean functions. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, Cambridge, MA, 2002. MIT Press.

[14] L. Kucera, A. Marchetti-Spaccamela, and M. Protassi. On learning monotone DNF formulae under uniform distributions. *Information and Computation*, 110:84–95, 1994.

[15] E. Kushilevitz and D. Roth. On learning visual concepts and DNF formulae. In *Proceedings of the Sixth Annual Conference on Computational Learning Theory*, pages 317–326, 1993.

[16] M. Minsky and S. Papert. *Perceptrons: an introduction to computational geometry.* MIT Press, Cambridge, MA, 1968.

[17] K. Sadohara. Learning of boolean functions using support vector machines. In *Proc. of the 12th International Conference on Algorithmic Learning Theory*, pages 106–118. Springer, 2001. LNAI 2225.

[18] Y. Sakai and A. Maruoka. Learning monotone log-term DNF formulas under the uniform distribution. *Theory of Computing Systems*, 33:17–33, 2000.

[19] R. Servedio. On PAC learning using winnow, perceptron, and a perceptron-like algorithm. In *Proceedings of the Twelfth Annual Conference on Computational Learning Theory*, pages 296–307, 1999.

[20] R. Servedio. On learning monotone DNF under product distributions. In *Proceedings of the Fourteenth Annual Conference on Computational Learning Theory*, pages 473–489, 2001.

[21] J. Shawe-Taylor and N. Cristianini. *An introduction to support vector machines.* Cambridge University Press, 2000.

[22] J. Tarui and T. Tsukiji. Learning DNF by approximating inclusion-exclusion formulae. In *Proceedings of the Fourteenth Conference on Computational Complexity*, pages 215–220, 1999.

[23] L. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[24] K. Verbeurgt. Learning DNF under the uniform distribution in quasi-polynomial time. In *Proceedings of the Third Annual Workshop on Computational Learning Theory*, pages 314–326, 1990.

[25] K. Verbeurgt. Learning sub-classes of monotone DNF on the uniform distribution. In *Proceedings of the Ninth Conference on Algorithmic Learning Theory*, pages 385–399, 1998.

[26] C. Watkins. Kernels from matching operations. Technical Report CSD-TR-98-07, Computer Science Department, Royal Holloway, University of London, 1999.

A Proof of Equation (4)

We must show that $\sqrt{\rho_k(u_1)} > 3m\left(\sqrt{\rho_k(.26n)} + \sqrt{\rho_k(u_1 - B)}\right)$. Since we are assuming that the sample S is $\mathcal{U}$-typical, we have $u_1 \geq .49n$ so $u_1 - B > 0.26n$. It thus suffices to show that $\rho_k(u_1) > 36m^2\rho_k(u_1 - B)$.

Case 1: $k \leq \frac{1}{2}(u_1 - B)$. Since $\rho_k(\ell) = \sum_{i=1}^{k}\binom{\ell}{i}$, for $k \leq \ell/2$ we have $\rho_k(\ell) \leq k\binom{\ell}{k}$. Also for all k, $\rho_k(\ell) \geq \binom{\ell}{k}$ so it suffices to show that

$$\binom{u_1}{k} > 36m^2 k\binom{u_1 - B}{k}.$$

This inequality is true if

$$\left(\frac{u_1}{u_1 - B}\right)^k > 36m^2 k.$$

Recall that $B = \frac{1}{66}\sqrt{\frac{n}{\log m}}$. Now using the fact that

$$\frac{u_1}{u_1 - B} = 1 + \frac{B}{u_1 - B} > 1 + \frac{B}{n} = 1 + \frac{1}{66\sqrt{n\log m}}$$

it suffices to show that

$$\left(1 + \frac{1}{66\sqrt{n\log m}}\right)^k > 36m^2 k.$$

Using the fact that $1 + x \geq e^{x/2}$ for $0 < x < 1$, we can see that this inequality holds if $k > 132\sqrt{n\log(m)}\ln(36m^2 n)$. Since $m = \mathrm{poly}(n)$, this is the case for $k = \omega(\sqrt{n}\log^{\frac{3}{2}} n)$.

Case 2: $\frac{1}{2}(u_1 - B) < k$. Since $\rho_k(u_1 - B) \leq 2^{u_1 - B}$, it suffices to show that

$$\sum_{i=1}^{\frac{u_1}{2} - \frac{B}{2}}\binom{u_1}{i} > 36m^2 \cdot 2^{u_1 - B}.$$

Since $\sqrt{u_1} > B/2$ it suffices to show that

$$\sum_{i=1}^{\frac{u_1}{2} - \sqrt{u_1}}\binom{u_1}{i} > 36m^2 \cdot 2^{u_1 - B}.$$

Standard binomial coefficient properties imply that the left side above is $\Theta(2^{u_1})$. Since $m = \mathrm{poly}(n)$ and $B = \frac{1}{66}\sqrt{\frac{n}{\log m}}$ this is greater than the right side.