Journal of Machine Learning Research 6 (2005) 1405–1429 Submitted 9/04; Revised 5/05; Published 9/05

Maximum Margin Algorithms with Boolean Kernels

Roni Khardon RONI@CS.TUFTS.EDU
Department of Computer Science
Tufts University
Medford, MA 02155, USA

Rocco A. Servedio ROCCO@CS.COLUMBIA.EDU
Department of Computer Science
Columbia University
New York, NY 10027, USA

Editor: Peter Bartlett

Abstract

Recent work has introduced Boolean kernels with which one can learn linear threshold functions 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 Probably Approximately Correct (PAC) learning model by 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 maximum margin algorithms using the Boolean kernels 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.

Keywords: computational learning theory, kernel methods, PAC learning, Boolean functions

1. Introduction

Maximum margin algorithms, notably the Support Vector Machines (SVM) introduced by Boser et al. (1992), have received considerable attention in recent years (see, e.g., Shawe-Taylor and Cristianini, 2000, 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 linear threshold 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 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 (see, e.g., Shawe-Taylor and Cristianini, 2000) 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., Friess et al., 1998; Gentile, 2001).

Both theoretical and experimental studies suggest that such algorithms may be able to take advantage of properties of the distribution and data to converge faster than what would be required by uniform convergence bounds. In particular, convergence bounds based on the maximum margin of the classifier on the observed data have been obtained by Shawe-Taylor et al. (1998) and by Shawe-Taylor and Cristianini (2000).

1.1 Can SVMs Learn DNF?

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 the Probably Approximately Correct (PAC) learning model of Valiant (1984) for learning Boolean formulas in Disjunctive Normal Form, or DNF. This question has been open since the introduction of the PAC model and has been intensively studied by many researchers (see, e.g., Blum et al., 1994; Blum and Rudich, 1995; Bshouty, 1996; Hancock and Mansour, 1991; Jackson, 1997; Khardon, 1994; Klivans and Servedio, 2001; Kucera et al., 1994; Kushilevitz and Roth, 1993; Sakai and Maruoka, 2000; Tarui and Tsukiji, 1999; Verbeurgt, 1990, 1998).

In this paper we analyze the performance of maximum margin algorithms when used with Boolean kernels to learn DNF formulas. Several authors including Khardon et al. (2002), Sadohara (2001), Watkins (1999) and Kowalczyk et al. (2002) 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 (2001); 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.

1.2 Discussion of the Problem and Previous Work

Recall that a polynomial size sample is sufficient for PAC learning any concept class where each concept in the class has a polynomial size description. In any such case, as shown by Blumer et al. (1987), an Occam algorithm which identifies a short consistent hypothesis in the class is a

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.

PAC learner. Thus the statistical ingredient of the problem of PAC learning polynomial size DNF expressions is in some sense solved and the main question seems to be computational. Yet, it is not known whether such an Occam algorithm exists.

As mentioned above recent work of Shawe-Taylor et al. (1998) and Shawe-Taylor and Cristianini (2000) has introduced convergence bounds for maximum margin learners. These bounds are independent of the dimension of the expanded feature space but they depend on the L_2 norm of examples in this space, as well as the margin obtained on the sample. In particular they depend on R/δ where δ is the margin and R bounds the L_2 norm of examples. It is instructive to consider applying these results in our setting, where we assume for concreteness that we are learning a function given by one k-monomial T, and that we are using the K_k monotone kernel with the maximum margin algorithm. The linear threshold representation for this function is $x_T \geq 1$, i.e. only one weight is non-zero and the (non-normalized) margin obtained is 1. However, the maximum L_2 norm of examples is $\Theta(n^{k/2})$ so the quantity R/δ is exponentially large. Seen in another way, we can normalize the examples to have a maximum norm of 1, but then the normalized margin obtained is $\Theta(n^{-k/2})$. Indeed, the bound given by Theorem 4.18 of the paper of Shawe-Taylor and Cristianini (2000) 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. As a result, using such bounds we cannot *a priori* conclude anything about the performance of the algorithm when it is run with a polynomial size sample.

Recently, several negative results have been obtained for embedding concept classes into Euclidean spaces (Ben-David et al., 2002; Forster et al., 2003). The results are best understood in terms of their relation to the convergence bounds. For example, Ben-David et al. (2002) show that there are concept classes for which there is no mapping into $[0,1]^N$ that achieves a large margin, for any N. This actually holds "for the majority of concept classes with low VC dimension". Other work of Forster et al. (2003) gives bounds on the margin (or the dimension required) for concrete concept classes. Again, the implication is that known convergence bounds do not imply success in these cases. It is worth noting that the notion of embedding used in these results is slightly stronger than the requirement in the upper bounds, in that the embedding and margin are for all the examples (or a large fraction of the instance space) and not just for a small sample. However, these results rule out any simple application of the upper bounds that use properties of the concept class directly.

Therefore, in many cases, and concretely in our case of learning DNF via the monomial kernel, the upper bounds provided by standard convergence theorems only imply that a large sample will guarantee successful generalization. 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. This is precisely the question studied in this paper. Notice the contrast with the discussion of Occam algorithms; here we have an efficient algorithm with no known bounds on hypothesis size. The question is whether its hypothesis provides a good generalization in a statistical sense.

The notion that this might succeed is not unreasonable. In an analogous situation, Servedio (1999) studied the generalization error of the Perceptron and Winnow algorithms for various problems. 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 showed that the Perceptron algorithm succeeds in polynomial time whereas the Winnow algorithm requires exponential time for the problems considered. Analogously, in this paper we perform detailed algorithm-specific analysis for the K_k kernel maximum margin algorithms.

In previous work we have studied a similar question with regard to the perceptron algorithm. In particular, Khardon et al. (2002) constructed a simple Boolean function and an example sequence for the online mistake-bound learning model, 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 $2^{\Omega(n)}$ many mistakes. 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 $1 \leq k \leq n$. We note here that maximum margin linear threshold learning algorithms are generally viewed as being more powerful than the simple Perceptron algorithm, and that PAC learning is generally viewed as being easier than online mistake bound learning (it is well known that any concept class which is efficiently learnable in the mistake bound model is efficiently PAC learnable, but the converse is not true as shown by Blum, 1994). Thus, the results of this work represent a substantial strengthening and generalization of the work of Khardon et al. (2002).

1.3 Our Results

In this paper we study the kernels corresponding to all monotone monomials of length up to k, which we denote by K_k. We also consider the polynomial kernel $K(x,y) = (x \cdot y)^k$, parametrized by the degree of the polynomial.

In addition to unaugmented maximum margin algorithms we also consider a natural scheme of structural risk minimization (SRM) that can be used with maximum margin algorithms over 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]

Combining either of these algorithms (i.e. with or without SRM scheme) with the monomial kernel we get a concrete and efficient algorithm that can be applied to the problem of learning DNF. We prove several negative results which establish strong limitations on the ability of such algorithms to learn DNF. Similar negative results are proved for the polynomial kernel as well.

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 $\varepsilon = \frac{1}{4 \cdot 2^{t(n)}}$. There is a $O(t(n)^{1/3})$-term monotone DNF over $t(n)$ relevant variables, and a distribution $\mathcal{D}$ over $\{0,1\}^n$ such that 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 $> \varepsilon$ for all k, and hence the final SRM hypothesis must also have error $> \varepsilon$.

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:

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.

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., Bshouty and Tamon, 1996; Servedio, 2001). 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 $\varepsilon = \frac{1}{4 \cdot 2^{t(n)}}$. There is a $O(t(n)^{1/3})$-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).

This result is representation based; we show that no possible hypothesis output by the K_k algorithm can have error less than ε. 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). 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.

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.

For $x, y \in \mathbb{R}^n$ we write $x \cdot y$ to denote the standard inner product $\sum_{i=1}^n x_i y_i$. Note that for $x, y \in \{0,1\}^n$, $x \cdot y$ calculates the number of bits which are 1 in both x and y. Our arguments will refer to L_1 and L_2 norms of vectors for which we use the notation $|x| = \sum |x_i|$ and $\|x\| = \sqrt{\sum x_i^2}$.

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 (see, e.g., Shawe-Taylor and Cristianini, 2000).

For a sample S and example $\langle x^i, 1\rangle$ in S we sometimes write $x^{i,+}$ to indicate that x^i is a positive example. Similarly $x^{j,-}$ is used to indicate that x^j is a negative example.

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 denote by $\phi(S)$ 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 takes as input a sample $S = \{\langle x^i, b_i\rangle\}_{i=1,\ldots,m}$ of m labeled examples.

  We assume that S contains both positive and negative examples, and that the sample is linearly separable. If these conditions do not hold then the maximum margin hypothesis is not defined. The assumptions are simply used to rule out the degenerate cases from the analysis.

  We also assume that $m = \text{poly}(n)$, i.e. that $m = n^{\Theta(1)}$ and we have both a lower and upper bound on the number of examples. The upper bound as usual limits the resources the algorithm uses. The lower bound is again simply used to rule out degenerate cases from the analysis.

- 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 $\|W\| = 1$.

SVM theory tells us that if $K(x,y)$ can be computed in $\text{poly}(n)$ time then the K-maximum margin learning algorithm runs in $\text{poly}(n,m) = \text{poly}(n)$ time and the output hypothesis $h(x)$ can be evaluated in $\text{poly}(n,m) = \text{poly}(n)$ time (see, e.g., Shawe-Taylor and Cristianini, 2000).

Our goal is to analyze the PAC learning ability of various kernel maximum margin learning algorithms. Recall (see, e.g., Kearns and Vazirani, 1994) 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 $\frac{1}{\delta}$, $\frac{1}{\varepsilon}$ 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 $\text{poly}(n)$. Given access to random labeled 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 $\Pr_{x \in \mathcal{D}}[h(x) \neq f(x)] \leq \varepsilon$. Applying this framework to the maximum margin learner, we assume that the sample S is drawn by taking IID samples from $\mathcal{D}$ and providing the label according to the target function f. 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 $|\phi_k(x)| = \rho_k(|x|)$.

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 4 *(Khardon, Roth, and Servedio, 2002) 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 $U\cdot\phi_k(x) \leq L\cdot\sqrt{N_2}$.

Finally we also use the following well-known tail bound on sums of independent random variables (see, e.g., Kearns and Vazirani, 1994):

Fact 2 (Chernoff Bounds) *Let $X_1,\ldots,X_m$ be a sequence of m independent 0/1-valued random variables, each of which has $E[X_i] = p$. Let X denote $\sum_{i=1}^{m} X_i$, so $E[X] = pm$. Then for $0 \leq \gamma \leq 1$, we have*

$$\Pr[X > (1+\gamma)pm] \leq e^{-mp\gamma^2/3} \quad \textit{and} \quad \Pr[X < (1-\gamma)pm] \leq e^{-mp\gamma^2/2}.$$

3. Distribution-Free Non-Learnability

We give a DNF and a distribution which are such that the maximum margin algorithm using 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. Let

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

where $4\ell^3 = t(n) = \log n$ so that the number of terms ℓ equals $\Theta(t(n)^{1/3}) = \Theta((\log n)^{1/3})$. 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 is -1 otherwise. The degree of the function is the degree of the polynomial p. A simple but useful observation is 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 (1968) (see also Klivans and Servedio, 2001) gave the following lower bound on polynomial threshold function degree for DNF:

Theorem 5 *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 6 *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 $\varepsilon = \frac{1}{4 \cdot 2^{t(n)}} = \frac{1}{4n}$.*

Proof If hypothesis h has error at most $\varepsilon = \frac{1}{4 \cdot 2^{t(n)}}$ 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 $\tau \le \frac{1}{2 \cdot 2^{t(n)}}$ 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 $\tau \le \frac{1}{2 \cdot 2^{t(n)}}$. ■

For larger values of k (in fact for all $k = \omega(1)$) we show that with high probability the maximum margin hypothesis will overfit the sample. We start by explaining the high level structure of the proof. Note that the target function depends on a small number of the features so most features are irrelevant for the target. On the other hand the distribution is constructed such that each example in the sample has a "large" weight on its own, whereas the weight of the common features in any two examples is "small". As a result of these facts, one can find a simple hypothesis with relatively large margin by using all the structure from the examples, i.e. fitting them exactly. Naturally such a hypothesis overfits the sample and provides little by way of generalizing to other examples. It is hard in general to analyze the maximum margin hypothesis directly, and in particular it does not necessarily follow the overfitting scheme of the simple hypothesis. However, our analysis uses the simple hypothesis to infer some properties of the maximum margin hypothesis and through this provide error bounds for it. The same structure is used again to analyze the polynomial kernel and for the analysis of the uniform distribution. However, the technical details underlying the analysis are different in each case.

The following definition captures typical properties of a sample from distribution $\mathcal{D}$:

Definition 7 *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 $0.99n^{2/3} \le |x| \le 1.01n^{2/3}$.*

- *Every pair of examples $x^{i,+}$ and $x^{j,-}$ in S satisfies $x^{i,+} \cdot x^{j,-} \leq 1.01n^{1/3}$.*

We can apply Chernoff bounds to analyze the second and third conditions in the definition (with $p = \frac{1}{n^{1/3}}$ and $p = \frac{1}{n^{2/3}}$ respectively) over the last $n - t(n) > n/2$ bits, and absorb the first $t(n)$ bits in the multiplicative (1 ± 0.01) divergence from the expected value in each case (recall that $t(n)$ is only $O(\log n)$). We thus have that the second and third conditions each fail with probability at most $2^{-n^{\Omega(1)}}$. Since the maximum margin algorithm uses m =poly$(n) = n^{\Omega(1)}$ many examples (see Section 2), the first condition fails with probability $2^{-m} = 2^{-n^{\Omega(1)}}$ as well. A union bound thus gives:

Lemma 8 *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 9 *Let S be a sample. The set $Z(S)$ consists of all positive examples $z \in \{0,1\}^n$ (i.e. $f(z) = 1$) which have the property that every example x in S satisfies $x \cdot z \leq 1.01n^{1/3}$.*

As above, we can apply Chernoff bounds with $p = \frac{1}{n^{2/3}}$ and use the union bound over all examples $x \in S$ to show that the probability that a random example z drawn from $\mathcal{D}$ will have $x \cdot z > 1.01n^{1/3}$ for any $x \in S$ is at most $2^{-n^{\Omega(1)}}$. Recall that f only depends on the first $t(n)$ bits and its terms are shorter than $t(n)$. Since the distribution is uniform over these bits we have $\Pr[f(z) = 1] \geq \frac{1}{2^{t(n)}} = \frac{1}{n}$. Thus, conditioning on z being a positive example we still have:

Lemma 10 *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 11 *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 satisfied in some positive example;
- $W'_T = 0$ if T is not satisfied 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, for any negative example x^- in the sample a term T contributes to $W' \cdot \phi(x^-)$ only if T is true in x^- and in some positive example. Now since x^- shares at most $1.01n^{1/3}$ ones with any positive example in the sample, the number of such terms is at most $m\rho_k(1.01n^{1/3})$. We therefore get $W' \cdot \phi(x^-) \leq m\rho_k(1.01n^{1/3})$. 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. ■

It is instructive to use a rough calculation and compare the margin obtained to the one calculated in the introduction. The main term in the bound above grows roughly as $\sqrt{\frac{\rho_k(n^{2/3})}{m}}$ which is exponentially larger than the constant value obtained by the correct classifier.

Lemma 12 *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 11. ■

Lemma 13 *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 $\varepsilon = \frac{1}{4 \cdot 2^{t(n)}} = \frac{1}{4n}$.*

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., Shawe-Taylor and Cristianini, 2000, Proposition 6.5) 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 8 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 12 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

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

We prove in Appendix A that this holds for any $k = \omega(1)$.

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 10 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

$(1-2^{-n^{\Omega(1)}})\frac{1}{2^{t(n)}} > \frac{1}{4\cdot 2^{t(n)}} = \frac{1}{4n}$ as claimed. ■

Together, Lemma 6 and Lemma 13 imply Result 1:

Theorem 14 *For any value of k, if the maximum margin algorithm (as defined in Section 2) 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 $\varepsilon = \frac{1}{4\cdot 2^{t(n)}} = \frac{1}{4n}$.*

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 $\mathcal{D}$ as follows. With probability $\frac{1}{4}$ the assignment 0^n is drawn. With probability $\frac{3}{4}$ we draw from $\mathcal{D}'$ where $x_1 = 1$ with probability $\frac{2}{3}$ and as before the other bits are 1 with probability $\frac{1}{n^{1/3}}$. Note that for the modified distribution the probability that $f(x) = 1$ is 0.5. It is easy to see the that previous arguments go through for this case and we get:

Theorem 15 *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 $\varepsilon = \frac{1}{2} - 2^{-n^{\Omega(1)}}$.*

Remark 16 The proofs above can be adapted to show the same non-learnability results for the polynomial kernel $K_k(x,y) = (x\cdot y)^k$ which is commonly being used with SVM systems. The low degree argument in Lemma 6 holds directly. We briefly sketch the ideas for the high degree case. First note that Lemmas 8 and 10 hold without modification. The argument in Lemma 11 does not go through if we use the same value of W' (since W' is defined in the expanded feature space and $\phi(x)$ is not a zero-one vector, it is not as easy to argue about the value of $W'\cdot\phi(x)$). However, we can use a simple modification to get a similar result. First note that for any $x \in \{0,1\}^n$, all features in $\phi(x)$ take only non-negative values. Now define W' to be $W' = \sum_{x^{i,+}\in S}\phi(x^{i,+})$. As in Lemma 11 we have:

- $W'\cdot\phi(x^+) = \sum_{x^{j,+}\in S}\phi(x^{j,+})\cdot\phi(x^+) \geq \phi(x^+)\cdot\phi(x^+) \geq (0.99n^{2/3})^k$ where the first inequality uses the fact that all features in the expanded space have a positive value and therefore all inner products in the sum are positive.

- $W'\cdot\phi(x^-) = \sum_{x^{j,+}\in S}\phi(x^{j,+})\cdot\phi(x^-) \leq m(1.01n^{1/3})^k$.

- $\|W'\| = \sqrt{(\sum_{x^{j,+}\in S}\phi(x^{j,+}))\cdot(\sum_{x^{i,+}\in S}\phi(x^{i,+}))} \leq \sqrt{m^2(1.01n^{2/3})^k}$.

So the maximum margin is at least

$$\frac{1}{2}\cdot\frac{(.99n^{2/3})^k - m(1.01n^{1/3})^k}{m\sqrt{(1.01n^{2/3})^k}}. \tag{4}$$

Now the proof of Lemma 12 shows that (4) is a lower bound on the threshold of the maximum margin classifier.

The argument in Lemma 13 needs to be changed since we need a bound on $W\cdot\phi(z)$. This can be derived as follows. Let U be such that $U_i \geq 0$ and $U_i = |W_i|$ so weights in U and W have the same magnitude but the weights in U are forced to be non-negative. Then we have that $\|U\| = \|W\| = 1$. For an example $z \in Z(S)$ we now have

$$\begin{aligned} W \cdot \phi(z) &\leq U \cdot \phi(z) \\ &\leq \sum_{x^i \in S} U \cdot \phi(z \cap x^i) \\ &\leq m(1.01 n^{1/3})^{k/2}. \end{aligned}$$

The first inequality holds since all entries in $\phi(z)$ are non-negative. The second inequality is true since both vectors do not have negative weights and a monomial contributes to $W \cdot \phi(z)$ only if it is true both in z and in at least one example in the sample (recall that, as in the proof of Lemma 13, the vector W is a linear combination of vectors $\phi(x) \in \phi(S)$). Therefore, each weight in $\phi(z)$ is represented by a weight in one of the intersections, and the value of the weight depends only on the monomial so it is the same in $\phi(z)$ and $\phi(z \cap x^i)$. Summing over all x_i in S gives an upper bound on the total contribution to $W \cdot \phi(z)$. The last inequality follows from the Cauchy-Schwarz inequality.

As a result of this upper bound on $W \cdot \phi(z)$, we have that z is misclassified if

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

This can be shown to hold for all $k = \omega(1)$.

4. Uniform Distribution

While Theorem 14 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.

It is easy to observe that the proof of Lemma 6 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.

The case of large k is more complex. In Section 3 we took advantage of the fact that 0^n occurred with high weight under the distribution $\mathcal{D}$. This provided a lower bound (of 0) on the value of $W \cdot \phi_k(x)$ for some negative example in the sample, and then we 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 explaining the idea we need some technical details.

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 labeled 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 $v_1 \leq v_2 \leq \cdots \leq v_{m^-}$.

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. This is captured by the following definition:

Definition 17 *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 $B = \frac{1}{66}\sqrt{\frac{n}{\log m}}$.*

Now, if the sample is positive skewed we can calculate a lower bound on $W \cdot \phi_k(x)$ for negative examples in the sample. The value of B is chosen so that this bound can be used to give a non-trivial bound for θ. The details of this argument are developed in Section 4.2. But we must first establish that the algorithm may indeed get a positive-skewed sample as input.

4.1 The Probability of Obtaining Positive-Skewed Samples

Theorem 18 *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.

Proof Our first step is to reduce to a situation in which the positive examples and negative examples are independent from each other.[3]

Let M_-, M_+ be any two positive integers. Consider the following new probabilistic experiment which we call E_{M_-,M_+}: first M_- draws are made from a binomial distribution $B(n-1, \frac{1}{2})$ to obtain (sorted) values $v_1 \leq \cdots \leq v_{M_-}$, and then M_+ draws are made from $1 + B(n-1, \frac{1}{2})$ to obtain (sorted) values $u_1 \leq \cdots \leq u_{M+}$. The values $v_1, \ldots, v_{M_-}$ are thus distributed identically to the weights of the negative examples in the scenario of Theorem 18 conditioned on $m_- = M_-$, and likewise for the $u_1, \ldots, u_{M_+}$ and the positive examples.

We define the following event:

- Event $A_{M_-,M+}$: $u_1 \geq v_1 + B$.

For succinctness let us write A_m for the event (in our original scenario of a size-m sample S drawn from $\mathcal{U}$) that S is positive-skewed. We then have

$$\begin{aligned}
\Pr[A_m] &\geq \Pr[.49m < m_-, m_+ < .51m] \cdot \Pr[A_m \mid .49m < m_-, m_+ < .51m] \\
&\geq (1 - 2^{-\Omega(m)}) \Pr[A_m \mid .49m < m_-, m_+ < .51m] \\
&\geq (1 - 2^{-\Omega(m)}) \min_{.49m < M_-, M_+ < .51m} \Pr[A_m \mid m_- = M_- \text{ and } m_+ = M_+] \\
&= (1 - 2^{-\Omega(m)}) \min_{.49m < M_-, M_+ < .51m} \Pr[A_{M_-,M_+}].
\end{aligned}$$

where the second inequality holds by Chernoff bound.

It thus suffices to show that for any values M_-, M_+ in $(.49m, .51m)$ we have $\Pr[A_{M_-,M_+}] \geq 0.0291$. Fix any M_-, M_+ in this range; we will henceforth only consider the experiment E_{M_-,M_+} in which any event involving only the u_i's is independent from any event involving only v_i's.

Let n' denote $n-1$. The idea of the next part of the proof is to show that with some probability v_1 falls into a relatively small left tail of the distribution while u_1 is bounded away from this tail. This gives us a gap between u_1 and v_1 as desired.

We consider u_1 first. For $1 \leq i \leq n'$ let $\psi(i)$ denote $\sum_{j=0}^{i-1} \binom{n'}{j} 2^{-n'}$. Note that $\psi(i)$ is precisely the weight in the "left tail up to i" of the distribution $1 + B(n', \frac{1}{2})$. Let X be the event that $\psi(u_1) \geq \frac{1}{2m}$

3. Note that this is not the case in S because the total number of examples is m so that more positive examples means less negative examples and vice versa. This dependence affects that probability over weights for the lightest positive and negative examples in a subtle way which is hard to analyze directly.

and $u_1 \leq n'/2$. In order to have $\psi(u_1) < \frac{1}{2m}$, at least one of the $M_+ < .51m$ draws from $1+B(n',\frac{1}{2})$ must land in the "left tail" of weight less than $\frac{1}{2m}$; by a union bound the probability that this occurs is less than $\frac{0.51}{2}$ and hence $\Pr[\psi(u_1) \geq \frac{1}{2m}] \geq 1 - \frac{0.51}{2} > 0.745$. The probability that $u_1 \geq n'/2$ is $2^{-\Omega(m)}$ and thus $\Pr[X] > 0.745 - 2^{-\Omega(m)} > 0.74$.

Next consider v_1. For $1 \leq i \leq n'$ let $\varphi(i)$ denote $\sum_{j=0}^{i} \binom{n'}{j} 2^{-n'}$; similar to $\psi(i)$ we have that $\varphi(i)$ captures the weight in the left tail of $B(n',\frac{1}{2})$. Let Y be the event that $\varphi_n(v_1) \leq \frac{1}{4m}$. This event fails to occur only if each of the M_- draws from $B(n',\frac{1}{2})$ misses the left tail of weight at most $\frac{1}{4m}$. We need to be slightly careful; note that $\varphi(\cdot)$ takes discrete values, so this tail may actually weigh less than $\frac{1}{4m}$ (e.g. conceivably $\varphi(22) = \frac{1}{m^2}$ and $\varphi(23) = \frac{1}{m}$.) To take care of this we will now show that this tail cannot weigh much less than $\frac{1}{4m}$.

For $c \geq 1$ let $\sigma(c)$ denote the largest integer such that $\varphi(\sigma(c)) \leq \frac{1}{cm}$.

Lemma 19 *For any constant $c \geq 1$ we have $\varphi(\sigma(c)) \geq \frac{1}{3cm}$.*

Proof Suppose not; then we have $\varphi(\sigma(c)) < \frac{1}{3cm}$ and $\varphi(\sigma(c)+1) > \frac{1}{cm}$. This implies that $\binom{n'}{\sigma(c)+1} > 2\sum_{j=0}^{\sigma(c)} \binom{n'}{j}$ so in particular $\binom{n'}{\sigma(c)+1} > 2\binom{n'}{\sigma(c)}$. This implies that $n' - \sigma(c) > 2\sigma(c) + 2$ which implies $\sigma(c) < (n'-2)/3$. But then Chernoff bound implies that for such values of $\sigma(c)$, $\varphi(\sigma(c)+1) = 2^{-\Omega(n')}$ which contradicts the inequality $\varphi(\sigma(c)+1) > \frac{1}{cm}$ since c is constant and m is polynomial in n. ■

The lemma implies that the left tail of weight at most $\frac{1}{4m}$ must have weight at least $\frac{1}{12m}$. Hence the probability that each of the $M_- > .49m$ draws from $B(n',\frac{1}{2})$ misses this left tail is at most $(1-\frac{1}{12m})^{.49m}$. This is at most 0.96 and hence $\Pr[Y] \geq 0.04$.

We next show that if events X and Y both occur then event A_{M_-,M_+} occurs. This will complete the proof of the theorem since the events X and Y are independent and we have that $\Pr[A_{M_-,M_+}] \geq \Pr[X]\Pr[Y] \geq 0.0296$.

Suppose, for the sake of contradiction, that events X and Y both occur but $u_1 \leq v_1 + (B-1)$. Since X occurs we have $\psi(u_1) \geq \frac{1}{2m}$, i.e.

$$\psi(u_1) = \sum_{j=0}^{u_1-1} \binom{n'}{j} 2^{-n'} \geq \frac{1}{2m}.$$

On the other hand since Y occurs we have $\varphi(v_1) \leq \frac{1}{4m}$, so

$$\sum_{j=0}^{v_1} \binom{n'}{j} 2^{-n'} \leq \frac{1}{4m}. \tag{5}$$

These two inequalities together clearly imply $u_1 > v_1$. In fact they imply

$$\sum_{j=v_1+1}^{u_1-1} \binom{n'}{j} 2^{-n'} \geq \frac{1}{4m}. \tag{6}$$

Thus we see that the weights between v_1+1 and u_1-1 have a substantial size. We next show that this implies that the weights below v_1 also have a substantial size, contradicting Equation (5). The following lemma is useful:

Lemma 20 *For all j such that $u_1 - 3B \leq j \leq u_1 - 1$ we have* $\binom{n'}{j} \geq \frac{1}{2}\binom{n'}{u_1-1}$.

Proof Clearly it suffices to prove that $2\binom{n'}{u_1-3B} \geq \binom{n'}{u_1-1}$. By event X we know that $\psi(u_1) \geq \frac{1}{2m}$. But the left tail Chernoff bound implies that unless

$$u_1 - 1 \geq \frac{n'}{2} - 2\sqrt{n' \log m} \tag{7}$$

we have $\psi(u_1) < \frac{1}{m^4} < \frac{1}{2m}$ so (7) must hold.

Let $c = \frac{n'}{2} - (u_1 - 1)$ so $0 < c \leq 2\sqrt{n' \log m}$. Now observe that for any b such that $b < 0.1n'$ we have

$$\frac{\binom{n'}{n'/2-b}}{\binom{n'}{n'/2-b-1}} = \frac{n'/2+b+1}{n'/2-b} = 1 + \frac{2b+1}{n'/2-b} < 1 + \frac{2b+1}{0.4n'} = 1 + \frac{5b+2.5}{n'}.$$

We thus have

$$\begin{aligned}
\frac{\binom{n'}{u_1-1}}{\binom{n'}{u_1-3B}} &= \frac{\binom{n'}{u_1-1}}{\binom{n'}{u_1-2}} \cdot \frac{\binom{n'}{u_1-2}}{\binom{n'}{u_1-3}} \cdot \cdots \cdot \frac{\binom{n'}{u_1-3B+1}}{\binom{n'}{u_1-3B}} \\
&= \frac{\binom{n'}{\frac{n'}{2}-c}}{\binom{n'}{\frac{n'}{2}-c-1}} \cdot \frac{\binom{n'}{\frac{n'}{2}-c-1}}{\binom{n'}{\frac{n'}{2}-c-2}} \cdot \cdots \cdot \frac{\binom{n'}{\frac{n'}{2}-(c+3B-2)}}{\binom{n'}{\frac{n'}{2}-(c+3B-3)}} \\
&< \left(1 + \frac{5c+2.5}{n'}\right)\left(1 + \frac{5(c+1)+2.5}{n'}\right) \cdots \left(1 + \frac{5(c+3B-2)+2.5}{n'}\right) \\
&< \left(1 + \frac{5(c+3B)+2.5}{n'}\right)^{3B} \\
&\leq e^{\frac{5(c+3B)+2.5}{n'}3B}
\end{aligned}$$

where we have used the inequality $1 + x \leq e^x$. The last quantity is at most $\sqrt{e} < 2$ provided that

$$3B < \frac{n'}{10(c+3B)+5}. \tag{8}$$

Now since $c \leq 2\sqrt{n' \log m}$ and we can bound $5 < \sqrt{n' \log m}$ and $30B < 0.5\sqrt{n' \log m}$ this holds if

$$3B = \frac{1}{22}\sqrt{\frac{n}{\log m}} < \frac{n'}{21.5\sqrt{n' \log m}} = \frac{1}{21.5}\sqrt{\frac{n'}{\log m}}$$

which is clearly true for sufficiently large n. ■

Recalling that $u_1 \leq v_1 + (B-1)$, we have that the sum in Equation (6) has at most $B-2$ terms. Now since event X holds, $u_1 < \frac{n'}{2}$ and therefore the largest of these terms is $\binom{n'}{u_1-1}2^{-n'}$. By Equation (6) we thus have that

$$\binom{n'}{u_1 - 1} 2^{-n'} \geq \frac{1}{4(B-2)m}. \tag{9}$$

Now Lemma 20 together with Equation (9), implies that we have

$$\sum_{j=u_1-3B}^{v_1-1}\binom{n'}{j}2^{-n'} > \sum_{j=u_1-3B}^{u_1-B-1}\binom{n'}{j}2^{-n'} \geq \sum_{j=u_1-3B}^{u_1-B-1}\frac{1}{2}\binom{n'}{u_1-1}2^{-n'} \geq \frac{2B}{2}\frac{1}{4(B-2)m} > \frac{1}{4m}$$

but this contradicts Equation (5). ■

4.2 Lower Bound for Large k

Using the fact that the sample is positive-skewed with constant probability we can prove the lower bound along the same lines as before.

Definition 21 *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 examples $x^{i,+}$ and $x^{j,-}$ in S satisfies $x^{i,+} \cdot x^{j,-} \leq 0.26n$.*

As above we can apply Chernoff bounds to derive the next two lemmas:

Lemma 22 *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 23 *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 24 *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 11:

Lemma 25 *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 $h'(x) = \text{sign}(W' \cdot \phi_k(x) - \theta')$ 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 $W' \cdot \phi_k(x^{j,-})$. ■

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

Lemma 26 *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 $W \cdot \phi_k(x^{1,-}) \geq -\sqrt{\rho_k(u_1 - B)}$. Now since $\theta \geq W \cdot \phi_k(x^{1,-}) + m_S$, Lemma 25 implies the result. ■

Putting all of the pieces together, we have:

Theorem 27 *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 $\varepsilon = \frac{1}{2} - 2^{-\Omega(n)}$.*

Proof By Lemma 22 and Theorem 18, the sample S used for learning is both $\mathcal{U}$-typical and positive skewed with probability at least $0.029 - 1/2^{-\Omega(n)}$ which is more than 0.028 for sufficiently large n. Consider any $z \in Z(S)$. Using the reasoning from Lemma 13, $W \cdot \phi(z)$ 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

$$\sqrt{\rho_k(u_1)} > 3m\left(\sqrt{\rho_k(.26n)} + \sqrt{\rho_k(u_1 - B)}\right). \tag{10}$$

Recall that $\rho_k(x) = \sum_{j=0}^{k}\binom{x}{j}$. Note that for $k = n$ (all-monomials kernel) the above inequality becomes $2^{u_1/2} > 3m\left(2^{.13n} + 2^{(u_1-B)/2}\right)$ which is clearly true. In Appendix B we show that Equation (10) 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 24 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 $\varepsilon = \frac{1}{2} - 2^{-\Omega(n)}$, and we are done. ■

Remark 28 Here again we can adapt the proofs to show non-learnability results for the polynomial kernel $K_k(x,y) = (x \cdot y)^k$. We modify the definition of W' in Lemma 25 as follows. For every positive example $x^{i,+}$ in the sample let $\hat{x}^{i,+}$ be the example obtained by picking an arbitrary subset of size u_1 of the original true bits and setting all other bits to 0. Now let $W' = \sum_{x^{i,+} \in S} \phi(\hat{x}^{i,+})$. Arguing as in Remark 16 we get that the maximum margin is at least

$$\frac{1}{2} \cdot \frac{u_1^k - m(0.26n)^k}{m\sqrt{u_1^k}}.$$

Now in Lemma 26 we get that $W' \cdot \phi(x^{1,-}) \geq -(u_1 - B)^{k/2}$ which again implies a lower bound on the threshold.

Finally, following Theorem 27 and the argument in Remark 16 one can show that for an example $z \in Z(S)$ we have $W \cdot \phi(z) \leq m\sqrt{(0.26n)^k}$ so that z is misclassified if

$$u_1^k - m(0.26n)^k - 2m\sqrt{u_1^k}\sqrt{(u_1 - B)^k} \geq 2m^2\sqrt{u_1^k(0.26n)^k}$$

which is true if

$$u_1^{k/2} > 5m^2(u_1 - B)^{k/2}.$$

Using the reasoning in Case 1 of Appendix B, one can show that this holds for $k = \omega(\sqrt{n}\log^{\frac{3}{2}} n)$.

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 the 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. Our results consider cases where the L_2 norm of examples in the expanded feature space is large. This seems necessary for learning DNF; note that while one can use an exponential function to define a kernel with weighted monomials where the weight decays exponentially depending on the degree k, this implies that the margin for functions of high degree is exponentially small.

While our results are negative there are several interesting avenues suggested by this work which may succeed; we discuss these briefly below. One direction 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 and kernel regularization. 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 (see, e.g., Verbeurgt, 1990) that when learning polynomial size DNF under the uniform distribution, 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 the uniform distribution 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 $f(x) = x_1$. 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.

Finally, the kernel we have used is natural in terms of capturing all monomials of a certain length but there are other ways to capture natural kernels for Boolean problems. An interesting possibility is using a kernel of parity functions and such a construction can indeed be given. The resulting representation is closely related to learning via the Fourier transform as done in the work of Linial et al. (1993); Kushilevitz and Mansour (1993); Mansour (1995) but the algorithmic ideas are very different to the ones used by maximum margin algorithms.

Acknowledgments

R.K.'s work was partly supported by NSF Grant IIS-0099446, and by a Research Semester Fellowship Award from Tufts University. Much of R.A.S.'s work was done while supported by a National Science Foundation Mathematical Sciences Postdoctoral Research Fellowship at Harvard University.

Appendix A. Proof of Equation (3)

To show that

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

it suffices to show that

$$\rho_k(.99n^{2/3}) > 3m\sqrt{\rho_k(1.01n^{1/3})\rho_k(1.01n^{2/3})}. \tag{11}$$

The proof uses several cases depending on the value of k relative to n.

Case 1: $k \le 0.505n^{1/3}$. Since $\rho_k(\ell) = \sum_{i=1}^{k} \binom{\ell}{i}$, for $k \le \ell/2$ we have that $\rho_k(\ell) \le k\binom{\ell}{k}$. For all k we have $\rho_k(\ell) \ge \binom{\ell}{k}$ so it suffices to show that

$$\binom{.99n^{2/3}}{k} > 3mk\sqrt{\binom{1.01n^{1/3}}{k}\binom{1.01n^{2/3}}{k}}$$

which is equivalent (clearing denominators from the binomial coefficients) to

$$\prod_{i=0}^{k-1}(.99n^{2/3}-i) > 3mk\sqrt{\prod_{i=0}^{k-1}(1.01n^{1/3}-i)(1.01n^{2/3}-i)}.$$

We now use the fact that for $i \geq 0$ we have $(A-i)(B-i) \leq (\sqrt{AB}-i)^2$ provided that $2\sqrt{AB} < A+B$; it is easy to see that this latter condition holds for $A = 1.01n^{1/3}$, $B = 1.01n^{2/3}$. It thus suffices to show that

$$\prod_{i=0}^{k-1}(.99n^{2/3}-i) > 3mk\prod_{i=0}^{k-1}(1.01n^{1/2}-i)$$

which in turn is implied by

$$\left(\frac{.99n^{2/3}}{1.01n^{1/2}}\right)^k > 3mn$$

(we used the fact that $k \leq n$ to obtain the right-hand side above). This holds as long as $k > \frac{\log(3mn)}{\log 0.98+\frac{1}{6}\log n} = \Theta(1)$ for any $m = \text{poly}(n)$. Therefore the condition holds for any $k = \omega(1)$.

Case 2: $0.5 \cdot 1.01n^{1/3} \leq k \leq 5 \cdot 1.01n^{1/3}$. In this case we use the bounds $(\frac{\ell}{k})^k \leq \rho_k(\ell) = \sum_{i=1}^{k}\binom{\ell}{i} \leq (\frac{e\ell}{k})^k$ for the first and third occurrences of ρ_k in equation (11) and we use $\rho_k(\ell) \leq 2^\ell$ for the second occurrence. It thus suffices to show that

$$\left(\frac{.99n^{2/3}}{k}\right)^k > 3m\sqrt{\left(\frac{e \cdot 1.01n^{2/3}}{k}\right)^k \cdot 2^{1.01n^{1/3}}}.$$

Applying the upper bound on k in the denominator on the left side, and the lower bound on k in the denominator on the right side, it suffices to show that

$$\left(\frac{.99}{5.05}n^{1/3}\right)^k > 3m\sqrt{\left(\frac{e \cdot 1.01}{0.505}n^{1/3}\right)^k \cdot 2^{1.01n^{1/3}}}$$

Now since $1.01n^{1/3} \leq 2k$ the condition holds if

$$\left(\frac{n^{1/3}}{6}\right)^k > 3m\left(2e \cdot n^{1/3}\right)^{k/2} \cdot 2^k$$

or equivalently if

$$\left(\frac{n^{1/6}}{12\sqrt{2e}}\right)^k > 3m.$$

This obviously holds since $k = \Theta(n^{1/3})$.

Case 3: $5 \cdot 1.01n^{1/3} \leq k \leq 0.25 \cdot 0.99n^{2/3}$. We use the same bounds as in the previous case to start the analysis, so we want to show that

$$\left(\frac{.99n^{2/3}}{k}\right)^k > 3m\sqrt{\left(\frac{e \cdot 1.01n^{2/3}}{k}\right)^k \cdot 2^{1.01n^{1/3}}}.$$

Since $1.01n^{1/3} \leq k/5$ it suffices to show that

$$\left(\frac{.99n^{2/3}}{k}\right)^k > 3m\left(\frac{e \cdot 1.01n^{2/3}}{k}\right)^{k/2} \cdot 2^{k/10}$$

which holds (taking k-th roots and rearranging) if and only if

$$\left(\frac{1}{2}\right)^{1/10} \cdot \frac{.99n^{2/3}}{k} \cdot \frac{\sqrt{k}}{n^{1/3}\sqrt{1.01 \cdot e}} = \left(\frac{1}{2}\right)^{1/10} \cdot \left(\frac{.99}{\sqrt{1.01 \cdot e}}\right) \cdot \frac{n^{1/3}}{\sqrt{k}} > (3m)^{1/k}.$$

Using our upper bound on k on the left side, the previous inequality holds if

$$(\frac{1}{2})^{1/10}\frac{.99}{\sqrt{1.01 \cdot e}} \cdot \frac{2}{\sqrt{.99}} > (3m)^{1/k}$$

and since the left side is greater than 1.1 the inequality holds if $k > \frac{\log 3m}{\log 1.1} = \Theta(\log n)$ for $m = \text{poly}(n)$. This obviously holds since $k = \Omega(n^{1/3})$.

Case 4: $0.25 \cdot 0.99n^{2/3} \leq k \leq 0.5 \cdot 0.99n^{2/3}$. We use the following bound (proved later) which holds for $0 < \alpha < 1$:

$$\sum_{i=1}^{\alpha q} \binom{q}{i} \geq \frac{1}{\sqrt{2\pi q}} 2^{H(\alpha)q} \tag{12}$$

where $H(p) = -p\log p - (1-p)\log(1-p)$ is the binary entropy function. Applying this bound to the left side of (11) with $q = .99n^{2/3}$ and $\alpha = k/q$, we have $.25 \leq \alpha \leq .5$ so $H(\alpha) > .81$. Since $\rho_k(\ell)$ is always at most 2^ℓ it suffices to show that

$$\frac{1}{\sqrt{2\pi \cdot .99n^{2/3}}} 2^{0.81 \cdot 0.99n^{2/3}} > 3m\sqrt{2^{1.01n^{2/3}+1.01n^{1/3}}}.$$

This is easily seen to hold for any $m = \text{poly}(n)$.

To prove the bound (12) we use Stirling's approximation $\sqrt{2\pi n}(\frac{n}{e})^n \leq n! \leq \sqrt{2\pi n}(\frac{n}{e})^n\sqrt{1+\frac{1}{2n}}$; in fact we use a weaker form with $\sqrt{2}$ instead of $\sqrt{1+\frac{1}{2n}}$ in the upper bound. We thus have

$$\begin{aligned}\sum_{i=1}^{\alpha q}\binom{q}{i} \geq \binom{q}{\alpha q} &= \frac{q!}{(\alpha q)!((1-\alpha)q)!} \geq \frac{\sqrt{2\pi q}}{2\sqrt{2\pi\alpha q}\sqrt{2\pi(1-\alpha)q}}\left(\frac{q}{e}\right)^q\left(\frac{e}{\alpha q}\right)^{\alpha q}\left(\frac{e}{(1-\alpha)q}\right)^{(1-\alpha)q} \\ &= \frac{1}{2\sqrt{2\pi\alpha(1-\alpha)q}}\alpha^{-\alpha q}(1-\alpha)^{-(1-\alpha)q} = \frac{1}{2\sqrt{2\pi\alpha(1-\alpha)q}}2^{qH(\alpha)}.\end{aligned}$$

Equation (12) follows since $\alpha(1-\alpha) \leq 1/4$.

Note that by using $\sum_{i=0}^{\alpha q}\binom{q}{i} \leq \alpha q\binom{q}{\alpha q}$ one can also obtain $\sum_{i=0}^{\alpha q}\binom{q}{i} \leq \frac{\sqrt{\alpha q}}{\sqrt{\pi(1-\alpha)}}2^{H(\alpha)q}$.

Case 5: $k \geq 0.5 \cdot 0.99n^{2/3}$. In this case we have $\rho_k(.99n^{2/3}) = \sum_{i=1}^{k}\binom{.99n^{2/3}}{i} \geq \frac{1}{2}2^{.99n^{2/3}}$. Thus it suffices to show that

$$\frac{1}{2} \cdot 2^{0.99n^{2/3}} > 3m\sqrt{2^{1.01n^{2/3}+1.01n^{1/3}}}$$

which is easily seen to hold for any $m = \text{poly}(n)$. Thus Equation (11) holds for all $k = \omega(1)$. ■

Appendix B. Proof of Equation (10)

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

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

This inequality is true if

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

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

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

Using the fact that $1 + x \geq e^{x/2}$ for $0 < x < 1$, we can see that this inequality holds if

$$k > 132\sqrt{n\log(m)}\ln(36m^2n).$$

Since $m = \text{poly}(n)$, 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 $\sqrt{u_1} > \sqrt{0.49n} > \frac{92}{132}\sqrt{n} > 92\frac{B}{2}$ it suffices to show that

$$\sum_{i=1}^{\frac{u_1}{2} - \frac{\sqrt{u_1}}{92}}\binom{u_1}{i} > 36m^2 \cdot 2^{u_1 - B}.$$

Using Stirling approximation it is easy to check that $\binom{q}{q/2} < \sqrt{1 + \frac{1}{2q}}\ \sqrt{\frac{2}{\pi}}\ \frac{1}{\sqrt{q}}\ 2^q$ and this implies that

$$\sum_{i=1}^{\frac{u_1}{2} - \frac{\sqrt{u_1}}{92}}\binom{u_1}{i} > \frac{1}{2}2^{u_1} - \frac{\sqrt{u_1}}{92}\sqrt{1 + \frac{1}{2u_1}}\ \sqrt{\frac{2}{\pi}}\frac{1}{\sqrt{u_1}}2^{u_1} > 0.49 \cdot 2^{u_1}$$

so the condition above holds if

$$0.49 \cdot 2^B > 36m^2.$$

This is clearly true since $m = \text{poly}(n)$ and $B = \frac{1}{66}\sqrt{\frac{n}{\log m}}$. ■

References

S. Ben-David, N. Eiron, and H.-U. Simon. Limitations of learning via embeddings in euclidean half spaces. *Journal of Machine Learning Research*, 3:441–461, 2002.

A. Blum. Separating distribution-free and mistake-bound learning models over the boolean domain. *SIAM Journal on Computing*, 23(5):990–1000, 1994.

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.

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.

A. Blumer, A. Ehrenfeucht, D. Haussler, and M. Warmuth. Occam's razor. *Information Processing Letters*, 24:377–380, 1987.

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.

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

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

J. Forster, N. Schmitt, H.-U. Simon, and T. Suttorp. Estimating the optimal margins of embeddings in euclidean half spaces. *Machine Learning*, 51(3):263–281, 2003.

T. Friess, N. Cristianini, and C. Campbell. The kernel adatron algorithm: a fast and simple learning procedure for support vector machine. In *Proceedings of the 15th International Conference on Machine Learning*, pages 188–196, 1998.

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

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.

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.

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

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

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.

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.

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.

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

E. Kushilevitz and Y. Mansour. Learning decision trees using the Fourier spectrum. *SIAM J. on Computing*, 22(6):1331–1348, 1993.

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.

N. Linial, Y. Mansour, and N. Nisan. Constant depth circuits, Fourier transform and learnability. *Journal of the ACM*, 40(3):607–620, 1993.

Y. Mansour. An $o(n^{\log\log n})$ learning algorithm for DNF under the uniform distribution. *Journal of Computer and System Sciences*, 50:543–550, 1995.

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

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.

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

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.

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

J. Shawe-Taylor, P. Bartlett, R. Williamson, and M. Anthony. Structural risk minimization over data-dependent hierarchies. *IEEE Transactions on Information Theory*, 44(5):1926–1940, 1998.

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

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.

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

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.

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.

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