Kernels as Features: On Kernels, Margins, and Low-dimensional Mappings*

Maria-Florina Balcan[1] and Avrim Blum[1] and Santosh Vempala[2]

[1] Computer Science Department, Carnegie Mellon University
{ninamf,avrim}@cs.cmu.edu
[2] Department of Mathematics, MIT
vempala@math.mit.edu

Abstract. Kernel functions are typically viewed as providing an implicit mapping of points into a high-dimensional space, with the ability to gain much of the power of that space without incurring a high cost if the result is linearly-separable by a large margin γ. However, the Johnson-Lindenstrauss lemma suggests that in the presence of a large margin, a kernel function can also be viewed as a mapping to a *low*-dimensional space, one of dimension only $\tilde{O}(1/\gamma^2)$. In this paper, we explore the question of whether one can efficiently produce such low-dimensional mappings, using only black-box access to a kernel function. That is, given just a program that computes $K(x, y)$ on inputs x, y of our choosing, can we efficiently construct an explicit (small) set of features that effectively capture the power of the implicit high-dimensional space? We answer this question in the affirmative if our method is also allowed black-box access to the underlying data distribution (i.e., unlabeled examples). We also give a lower bound, showing that if we do not have access to the distribution, then this is not possible for an *arbitrary* black-box kernel function; we leave as an open problem, however, whether this can be done for standard kernel functions such as the polynomial kernel.
Our positive result can be viewed as saying that designing a good kernel function is much like designing a good feature space. Given a kernel, by running it in a black-box manner on random unlabeled examples, we can *efficiently* generate an explicit set of $\tilde{O}(1/\gamma^2)$ features, such that if the data was linearly separable with margin γ under the kernel, then it is approximately separable in this new feature space.

1 Introduction

Kernels functions have become a powerful tool in Machine Learning [8, 9, 15, 18, 24, 20, 22, 23, 25, 26]. A kernel function can be viewed as allowing one to implicitly map data into a high-dimensional space and to perform certain operations there without paying a high price computationally. Furthermore, if the data has a large margin linear separator in that space, then one can avoid paying a high price in terms of sample size as well [3, 21, 11].

* A preliminary version of this paper appeared in Proceedings of the 15th International Conference on Algorithmic Learning Theory. Springer LNAI 3244, pp. 194-205, 2004.

The starting point for this paper is the observation that if a learning problem indeed has the large margin property under some kernel $K(x, y) = \phi(x) \cdot \phi(y)$, then by the Johnson-Lindenstrauss lemma, a *random* linear projection of the "ϕ-space" down to a *low* dimensional space approximately preserves linear separability [1, 2, 10, 16]. Specifically, suppose data comes from some underlying distribution D over the input space X and is labeled by some target function c. If D is such that the target function has margin γ in the ϕ-space,[3] then a random linear projection of the ϕ-space down to a space of dimension $d = O\left(\frac{1}{\gamma^2} \log \frac{1}{\varepsilon\delta}\right)$ will, with probability at least $1 - \delta$, have a linear separator with error rate at most ε (see Arriaga and Vempala [2] and also Theorem 3 of this paper). This means that for any kernel K and margin γ, we can, in principle, think of K as mapping the input space X into an $\tilde{O}(1/\gamma^2)$-dimensional space, in essence serving as a method for representing the data in a new (and not too large) feature space.

The question we consider in this paper is whether, given kernel K, we can in fact produce such a mapping efficiently. The problem with the above observation is that it requires explicitly computing the function $\phi(x)$. In particular, the mapping of X into R^d that results from applying the Johnson-Lindenstrauss lemma is a function $F(x) = (r_1 \cdot \phi(x), \ldots, r_d \cdot \phi(x))$, where $r_1, \ldots, r_d$ are random vectors in the ϕ-space. Since for a given kernel K, the dimensionality of the ϕ-space might be quite large, this is not efficient. Instead, what we would like is an efficient procedure that given $K(.,.)$ as a black-box program, produces a mapping with the desired properties and with running time that depends (polynomially) only on $1/\gamma$ and the time to compute the kernel function K, with no dependence on the dimensionality of the ϕ-space.

Our main result is a positive answer to this question, if our procedure for computing the mapping is also given black-box access to the distribution D (i.e., unlabeled data). Specifically, given black-box access to a kernel function $K(x, y)$, a margin value γ, access to unlabeled examples from distribution D, and parameters ε and δ, we can in polynomial time construct a mapping $F : X \to R^d$ (i.e., to a set of d real-valued features) where $d = O\left(\frac{1}{\gamma^2} \log \frac{1}{\varepsilon\delta}\right)$ with the following property. If the target concept indeed has margin γ in the ϕ-space, then with probability $1 - \delta$ (over randomization in our choice of mapping function), the induced distribution in R^d is separable with error $\leq \varepsilon$. In fact, not only will the data in R^d be separable, but it will be separable with margin $\Omega(\gamma)$. Note that the logarithmic dependence on ε implies that if the learning problem has a perfect separator of margin γ in the ϕ-space, we can set ε small enough so that with high probability a set S of $O(d \log d)$ labeled examples would be perfectly separable in the mapped space. This means we could apply an arbitrary zero-noise linear-separator learning algorithm in the mapped space, such as a highly-optimized

[3] That is, there exists a linear separator in the ϕ-space such that any example from D is correctly classified by margin γ. See Section 2 for formal definitions. In Section 4.1 we consider the more general case that only a $1 - \alpha$ fraction of the distribution D is separated by margin γ.

linear-programming package. However, while the dimension d has a logarithmic dependence on $1/\varepsilon$, the number of (unlabeled) examples we use to produce our mapping is $\tilde{O}(1/(\gamma^2\varepsilon))$.

To give a feel of what such a mapping might look like, suppose we are willing to use dimension $d = O(\frac{1}{\varepsilon}[\frac{1}{\gamma^2} + \ln\frac{1}{\delta}])$ (so this is linear in $1/\varepsilon$ rather than logarithmic) and we are not concerned with preserving margins and only want approximate separability. Then we show the following especially simple procedure suffices. Just draw a random sample of d unlabeled points $x_1, \ldots, x_d$ from D and define $F(x) = (K(x, x_1), \ldots, K(x, x_d))$. That is, if we think of K not so much as an implicit mapping into a high-dimensional space but just as a similarity function over examples, what we are doing is drawing d "reference" points and then defining the ith feature of x to be its similarity with reference point i. We show (Corollary 1) that under the assumption that the target function has margin γ in the ϕ space, with high probability the data will be approximately separable under this mapping. Thus, this gives a particularly simple way of using the kernel and unlabeled data for feature generation.

Given the above results, a natural question is whether it might be possible to perform mappings of this type without access to the underlying distribution. In Section 5 we show that this is in general *not* possible, given only black-box access (and polynomially-many queries) to an *arbitrary* kernel K. However, it may well be possible for specific standard kernels such as the polynomial kernel or the gaussian kernel.

Relation to Support Vector Machines and Margin Bounds: Given a set S of n training examples, the kernel matrix defined over S can be viewed as placing S into an n-dimensional space, and the weight-vector found by an SVM will lie in this space and maximize the margin with respect to the training data. Our goal is to define a mapping over the entire distribution, with guarantees with respect to the distribution itself. In addition, the construction of our mapping requires only unlabeled examples, and so could be performed before seeing any labeled training data if unlabeled examples are freely available. There is, however, a close relation to margin bounds [21, 3] for SVMs (see Remark 1 in Section 3), though the dimension of our output space is lower than that produced by combining SVMs with standard margin bounds.

Our goals are to some extent related to those of Ben-David et al. [4, 5]. They show negative results giving simple classes of learning problems for which one cannot construct a mapping to a low-dimensional space under which all functions in the class are linearly separable. We restrict ourselves to situations where we know that such mappings exist, but our goal is to produce them efficiently.

Interpretation: Kernel functions are often viewed as providing much of the power of an implicit high-dimensional space without having to pay for it. Our results suggest that an alternative view of kernels is as a (distribution-dependent) mapping into a low-dimensional space. In this view, designing a good kernel function is much like designing a good feature space. Given a kernel, by running it in a black-box manner on random unlabeled examples, one can efficiently generate an explicit set of $\tilde{O}(1/\gamma^2)$ features, such that if the data was linearly separable with margin γ under the kernel, then it is approximately separable using these new features.

Outline of this paper: We begin with by giving our formal model and definitions in Section 2. We then in Section 3 show that the simple mapping described earlier in this section preserves approximate separability, and give a modification that approximately preserves both separability and margin. Both of these map data into a d-dimensional space for $d = O(\frac{1}{\varepsilon}[\frac{1}{\gamma^2} + \ln \frac{1}{\delta}])$. In Section 4, we give an improved mapping, that maps data to a space of dimension only $O(\frac{1}{\gamma^2} \log \frac{1}{\varepsilon\delta})$. This logarithmic dependence on $\frac{1}{\varepsilon}$ means we can set ε small enough as a function of the dimension and our input error parameter that we can then plug in a generic zero-noise linear separator algorithm in the mapped space (assuming the target function was perfectly separable with margin γ in the ϕ-space). In Section 5 we give a lower bound, showing that for a black-box kernel, one must have access to the underlying distribution D if one wishes to produce a good mapping into a low-dimensional space. In Section 6 we present experimental results using our mappings on both synthetic and standard datasets, and finally we end with a short discussion in Section 7.

2 Notation and Definitions

We assume that data is drawn from some distribution D over an instance space X and labeled by some unknown target function $c : X \to \{-1, +1\}$. We use P to denote the combined distribution over labeled examples.

A *kernel* K is a pairwise function $K(x, y)$ that can be viewed as a "legal" definition of inner product. Specifically, there must exist a function ϕ mapping X into a possibly high-dimensional Euclidean space such that $K(x, y) = \phi(x) \cdot \phi(y)$. We call the range of ϕ the "ϕ-space", and use $\phi(D)$ to denote the induced distribution in the ϕ-space produced by choosing random x from D and then applying $\phi(x)$.

We say that for a set S of labeled examples, a vector w in the ϕ-space has margin γ if:

$$\min_{(x,\ell)\in S} \left[\ell \frac{w \cdot \phi(x)}{||w||\ ||\phi(x)||} \right] \geq \gamma.$$

That is, w has margin γ if any labeled example in S is correctly classified by the linear separator $w \cdot \phi(x) \geq 0$, and furthermore the cosine of the angle between w and $\phi(x)$ has magnitude at least γ.[4] If such a vector w exists, then we say that S is linearly separable with margin γ under the kernel K. For simplicity, we are

[4] Often margin is defined without normalizing by the length of the examples, though in that case the "γ^2" term in sample complexity bounds becomes "γ^2/R^2", where R is the maximum $||\phi(x)||$ over $x \in S$. Technically, normalizing produces a stronger bound because we are taking the minimum of a ratio, rather than the ratio of a minimum to a maximum.

only considering separators that pass through the origin, though our results can be adapted to the general case as well (see Section 4.1).

We can similarly talk in terms of the distribution P rather than a sample S. We say that a vector w in the ϕ-space has margin γ with respect to P if:

$$\Pr_{(x,\ell)\leftarrow P}\left[\ell\frac{w\cdot\phi(x)}{||w||\ ||\phi(x)||}<\gamma\right]=0.$$

If such a vector w exists, then we say that P is linearly separable with margin γ under K (or just that P has margin γ in the ϕ-space). One can also weaken the notion of perfect separability. We say that a vector w in the ϕ-space has error α at margin γ if:

$$\Pr_{(x,\ell)\leftarrow P}\left[\ell\frac{w\cdot\phi(x)}{||w||\ ||\phi(x)||}<\gamma\right]\leq\alpha.$$

Our starting assumption in this paper will be that P is perfectly separable with margin γ under K, but we can also weaken the assumption to the existence of a vector w with error α at margin γ, with a corresponding weakening of the implications (see Section 4.1). Our goal is a mapping $F : X \to R^d$ where d is not too large that approximately preserves separability, and, ideally, the margin. We use $F(D)$ to denote the induced distribution in R^d produced by selecting points in X from D and then applying F, and use $F(P) = F(D, c)$ to denote the induced distribution on labeled examples.

For a set of vectors $v_1, v_2, \ldots, v_k$ in Euclidean space, let $\mathrm{span}(v_1, \ldots, v_k)$ denote the set of vectors v that can be written as a linear combination $a_1v_1 + \ldots + a_kv_k$. Also, for a vector v and a subspace Y, let $\mathrm{proj}(v, Y)$ be the orthogonal projection of v down to Y. So, for instance, $\mathrm{proj}(v, \mathrm{span}(v_1, \ldots, v_k))$ is the orthogonal projection of v down to the space spanned by $v_1, \ldots, v_k$. We note that given a set of vectors $v_1, \ldots, v_k$ and the ability to compute dot-products, this projection can be computed efficiently by solving a set of linear equalities.

3 Two simple mappings

Our goal is a procedure that given black-box access to a kernel function $K(.,.)$, unlabeled examples from distribution D, and a margin value γ, produces a (probability distribution over) mappings $F : X \to R^d$ with the following property: if the target function indeed has margin γ in the ϕ-space, then with high probability our mapping will approximately preserve linear separability. In this section, we analyze two methods that both produce a space of dimension $d = O(\frac{1}{\varepsilon}[\frac{1}{\gamma^2} + \ln\frac{1}{\delta}])$, where ε is our desired bound on the error rate of the best separator in the mapped space. The second of these mappings in fact satisfies a stronger condition that its output will be approximately separable at margin $\gamma/2$ (rather than just approximately separable). This property will allow us to use this mapping as a first step in a better mapping in Section 4.

The following lemma is key to our analysis.

Lemma 1. *Consider any distribution over labeled examples in Euclidean space such that there exists a vector w with margin γ. Then if we draw*

$$d \geq \frac{8}{\varepsilon}\left[\frac{1}{\gamma^2} + \ln\frac{1}{\delta}\right]$$

examples $z_1, \ldots, z_d$ i.i.d. from this distribution, with probability $\geq 1-\delta$, there exists a vector w' in $\mathrm{span}(z_1, \ldots, z_d)$ that has error at most ε at margin $\gamma/2$.

Remark 1. Before proving Lemma 1, we remark that a somewhat weaker bound on d can be derived from the machinery of margin bounds. Margin bounds [21, 3] tell us that using $d = O(\frac{1}{\varepsilon}[\frac{1}{\gamma^2}\log^2(\frac{1}{\gamma\varepsilon}) + \log\frac{1}{\delta}])$ points, with probability $1-\delta$, *any* separator with margin $\geq \gamma$ over the observed data has true error $\leq \varepsilon$. Thus, the projection of the target function w into the space spanned by the observed data will have true error $\leq \varepsilon$ as well. (Projecting w into this space maintains the value of $w \cdot z_i$, while possibly shrinking the vector w, which can only increase the margin over the observed data.) The only technical issue is that we want as a conclusion for the separator not only to have a low error rate over the distribution, but also to have a large margin. However, this can be obtained from the double-sample argument used in [21, 3] by using a $\gamma/4$-cover instead of a $\gamma/2$-cover. Margin bounds, however, are a bit of an overkill for our needs, since we are only asking for an existential statement (the *existence* of w') and not a universal statement about all separators with large empirical margins. For this reason we are able to get a better bound by a direct argument from first principles.

Proof (Lemma 1). For any set of points S, let $w_{in}(S)$ be the projection of w to $\mathrm{span}(S)$, and let $w_{out}(S)$ be the orthogonal portion of w, so that $w = w_{in}(S) + w_{out}(S)$ and $w_{in}(S) \perp w_{out}(S)$. Also, for convenience, assume w and all examples z are unit-length vectors (since we have defined margins in terms of angles, we can do this without loss of generality). Now, let us make the following definitions. Say that $w_{out}(S)$ is *large* if $\Pr_z(|w_{out}(S) \cdot z| > \gamma/2) \geq \varepsilon$, and otherwise say that $w_{out}(S)$ is *small*. Notice that if $w_{out}(S)$ is small, we are done, because $w \cdot z = (w_{in}(S) \cdot z) + (w_{out}(S) \cdot z)$, which means that $w_{in}(S)$ has the properties we want. That is, there is at most an ε probability mass of points z whose dot-product with w and $w_{in}(S)$ differ by more than $\gamma/2$. So, we need only to consider what happens when $w_{out}(S)$ is large.

The crux of the proof now is that if $w_{out}(S)$ is large, this means that a new random point z has at least an ε chance of significantly improving the set S. Specifically, consider z such that $|w_{out}(S) \cdot z| > \gamma/2$. Let $z_{in}(S)$ be the projection of z to $\mathrm{span}(S)$, let $z_{out}(S) = z - z_{in}(S)$ be the portion of z orthogonal to $\mathrm{span}(S)$, and let $z' = z_{out}(S)/||z_{out}(S)||$. Now, for $S' = S \cup \{z\}$, we have $w_{out}(S') = w_{out}(S) - \mathrm{proj}(w_{out}(S), \mathrm{span}(S')) = w_{out}(S) - (w_{out}(S) \cdot z')z'$, where the last equality holds because $w_{out}(S)$ is orthogonal to $\mathrm{span}(S)$ and so its projection onto $\mathrm{span}(S')$ is the same as its projection onto z'. Finally, since $w_{out}(S')$ is orthogonal to z' we have $||w_{out}(S')||^2 = ||w_{out}(S)||^2 - |w_{out}(S) \cdot z'|^2$, and since $|w_{out}(S) \cdot z'| \geq |w_{out}(S) \cdot z_{out}(S)| = |w_{out}(S) \cdot z|$, this implies by definition of z that $||w_{out}(S')||^2 < ||w_{out}(S)||^2 - (\gamma/2)^2$.

So, we have a situation where so long as w_{out} is large, each example has at least an ε chance of reducing $||w_{out}||^2$ by at least $\gamma^2/4$, and since $||w||^2 = ||w_{out}(\emptyset)||^2 = 1$, this can happen at most $4/\gamma^2$ times. Chernoff bounds state that a coin of bias ε flipped $n = \frac{8}{\varepsilon}\left[\frac{1}{\gamma^2} + \ln\frac{1}{\delta}\right]$ times will with probability $1-\delta$ have at least $n\varepsilon/2 \geq 4/\gamma^2$ heads. Together, these imply that with probability at least $1-\delta$, $w_{out}(S)$ will be small for $|S| \geq \frac{8}{\varepsilon}\left[\frac{1}{\gamma^2} + \ln\frac{1}{\delta}\right]$ as desired. □

Lemma 1 implies that if P is linearly separable with margin γ under K, and we draw $d = \frac{8}{\varepsilon}[\frac{1}{\gamma^2} + \ln\frac{1}{\delta}]$ random unlabeled examples $x_1, \ldots, x_d$ from D, then with probability at least $1-\delta$ there is a separator w' in the ϕ-space with error rate at most ε that can be written as

$$w' = \alpha_1\phi(x_1) + \ldots + \alpha_d\phi(x_d).$$

Notice that since $w' \cdot \phi(x) = \alpha_1 K(x, x_1) + \ldots + \alpha_d K(x, x_d)$, an immediate implication is that if we simply think of $K(x, x_i)$ as the ith "feature" of x — that is, if we define $F_1(x) = (K(x, x_1), \ldots, K(x, x_d))$ — then with high probability the vector $(\alpha_1, \ldots, \alpha_d)$ is an approximate linear separator of $F_1(P)$. So, the kernel and distribution together give us a particularly simple way of performing feature generation that preserves (approximate) separability. Formally, we have the following.

Corollary 1. *If P has margin γ in the ϕ-space, then with probability $\geq 1-\delta$, if $x_1, \ldots, x_d$ are drawn from D for $d = \frac{8}{\varepsilon}\left[\frac{1}{\gamma^2} + \ln\frac{1}{\delta}\right]$, the mapping*

$$F_1(x) = (K(x, x_1), \ldots, K(x, x_d))$$

produces a distribution $F_1(P)$ that is linearly separable with error at most ε.

Unfortunately, the above mapping F_1 may not preserve margins because we do not have a good bound on the length of the vector $(\alpha_1, \ldots, \alpha_d)$ defining the separator in the new space, or the length of the examples $F_1(x)$. The key problem is that if many of the $\phi(x_i)$ are very similar, then their associated features $K(x, x_i)$ will be highly correlated. Instead, to preserve margin we want to choose an orthonormal basis of the space spanned by the $\phi(x_i)$: i.e., to do an orthogonal projection of $\phi(x)$ into this space. Specifically, let $S = \{x_1, ..., x_d\}$ be a set of of $\frac{8}{\varepsilon}[\frac{1}{\gamma^2} + \ln\frac{1}{\delta}]$ unlabeled examples from D. We can then implement the desired orthogonal projection of $\phi(x)$ as follows. Run $K(x, y)$ for all pairs $x, y \in S$, and let $M(S) = (K(x_i, x_j))_{x_i, x_j \in S}$ be the resulting kernel matrix. Now decompose $M(S)$ into $U^T U$, where U is an upper-triangular matrix. Finally, define the mapping $F_2 : X \to R^d$ to be $F_2(x) = F_1(x)U^{-1}$, where F_1 is the mapping of Corollary 1. This is equivalent to an orthogonal projection of $\phi(x)$ into span$(\phi(x_1), \ldots, \phi(x_d))$. Technically, if U is not full rank then we want to use the (Moore-Penrose) pseudoinverse [6] of U in place of U^{-1}.

We now claim that by Lemma 1, this mapping F_2 maintains approximate separability at margin $\gamma/2$.

Theorem 1. *If P has margin γ in the ϕ-space, then with probability $\geq 1-\delta$, the mapping $F_2 : X \to R^d$ for $d \geq \frac{8}{\varepsilon}\left[\frac{1}{\gamma^2} + \ln\frac{1}{\delta}\right]$ has the property that $F_2(P)$ is linearly separable with error at most ε at margin $\gamma/2$.*

Proof. The theorem follows directly from Lemma 1 and the fact that F_2 is an orthogonal projection. Specifically, since $\phi(D)$ is separable at margin γ, Lemma 1 implies that for $d \geq \frac{8}{\varepsilon}\left[\frac{1}{\gamma^2} + \ln\frac{1}{\delta}\right]$, with probability at least $1-\delta$, there exists a vector w' that can be written as $w' = \alpha_1\phi(x_1) + ... + \alpha_d\phi(x_d)$, that has error at most ε at margin $\gamma/2$ with respect to $\phi(P)$, i.e.,

$$\Pr_{(x,\ell)\leftarrow P}\left[\frac{\ell(w'\cdot\phi(x))}{||w'||\ ||\phi(x)||} < \frac{\gamma}{2}\right] \leq \varepsilon.$$

Now consider $\overline{w} = \alpha_1 F_2(x_1) + \ldots + \alpha_d F_2(x_d)$. Since F_2 is an orthogonal projection and the $\phi(x_i)$ are clearly already in the space spanned by the $\phi(x_i)$, $\overline{w}$ can be viewed as the same as w' but just written in a different basis. In particular, we have $||\overline{w}|| = ||w'||$, and $w'\cdot\phi(x) = \overline{w}\cdot F_2(x)$ for all $x \in X$. Since $||F_2(x)|| \leq ||\phi(x)||$ for every $x \in X$, we get that $\overline{w}$ has error at most ε at margin $\gamma/2$ with respect to $F_2(P)$, i.e.,

$$\Pr_{(x,\ell)\leftarrow P}\left[\frac{\ell(\overline{w}\cdot F_2(x))}{||\overline{w}||\ ||F_2(x)||} < \frac{\gamma}{2}\right] \leq \varepsilon.$$

Therefore, for our choice of d, with probability at least $1-\delta$ (over randomization in our choice of F_2), there exists a vector $\overline{w} \in R^d$ that has error at most ε at margin $\gamma/2$ with respect to $F_2(P)$. □

Notice that the running time to compute $F_2(x)$ is polynomial in $1/\gamma, 1/\varepsilon, 1/\delta$ and the time to compute the kernel function K.

4 An improved mapping

We now describe an improved mapping, in which the dimension d has only a logarithmic, rather than linear, dependence on $1/\varepsilon$. The idea is to perform a two-stage process, composing the mapping from the previous section with a random linear projection from the range of that mapping down to the desired space. Thus, this mapping can be thought of as combining two types of random projection: a projection based on points chosen at random from D, and a projection based on choosing points uniformly at random in the intermediate space.

We begin by stating a result from [1, 2, 10, 14, 16] that we will use. Here $N(0,1)$ is the standard Normal distribution with mean 0 and variance 1 and $U(-1,1)$ is the distribution that has probability 1/2 on -1 and probability 1/2 on 1. Here we present the specific form given in [2].

Theorem 2 (Neuronal RP [2]). *Let $u, v \in R^n$. Let $u' = \frac{1}{\sqrt{k}}uA$ and $v' = \frac{1}{\sqrt{k}}vA$ where A is a $n \times k$ random matrix whose entries are chosen independently from either $N(0,1)$ or $U(-1,1)$. Then,*

$$\Pr_A\left[(1-\varepsilon)||u-v||^2 \leq ||u'-v'||^2 \leq (1+\varepsilon)||u-v||^2\right] \geq 1 - 2e^{-(\varepsilon^2-\varepsilon^3)\frac{k}{4}}.$$

Let $F_2 : X \to R^{d_2}$ be the mapping from Section 3 using $\varepsilon/2$ and $\delta/2$ as its error and confidence parameters respectively. Let $\hat{F} : R^{d_2} \to R^{d_3}$ be a random projection as in Theorem 2. Specifically, we pick A to be a random $d_2 \times d_3$ matrix whose entries are chosen i.i.d. $N(0,1)$ or $U(-1,1)$. We then set $\hat{F}(x) = \frac{1}{\sqrt{d_3}} xA$. We finally consider our overall mapping $F_3 : X \to R^{d_3}$ to be $F_3(x) = \hat{F}(F_2(x))$.

We now claim that for $d_2 = O(\frac{1}{\varepsilon}[\frac{1}{\gamma^2} + \ln \frac{1}{\delta}])$ and $d_3 = O(\frac{1}{\gamma^2}\log(\frac{1}{\varepsilon\delta}))$, with high probability, this mapping has the desired properties. The basic argument is that the initial mapping F_2 maintains approximate separability at margin $\gamma/2$ by Lemma 1, and then the second mapping approximately preserves this property by Theorem 2.

Theorem 3. *If P has margin γ in the ϕ-space, then with probability at least $1-\delta$, the mapping $F_3 = \hat{F} \circ F_2 : X \to R^{d_3}$, for values $d_2 = O\left(\frac{1}{\varepsilon}\left[\frac{1}{\gamma^2} + \ln \frac{1}{\delta}\right]\right)$ and $d_3 = O\left(\frac{1}{\gamma^2}\log(\frac{1}{\varepsilon\delta})\right)$, has the property that $F_3(P)$ is linearly separable with error at most ε at margin $\gamma/4$.*

Proof. By Lemma 1, with probability at least $1-\delta/2$ there exists a separator w in the intermediate space R^{d_2} with error at most $\varepsilon/2$ at margin $\gamma/2$. Let us assume this in fact occurs. Now, consider some point $x \in R^{d_2}$. Theorem 2 implies that a choice of $d_3 = O(\frac{1}{\gamma^2}\log(\frac{1}{\varepsilon\delta}))$ is sufficient so that under the random projection $\hat{F}$, with probability at least $1-\varepsilon\delta/4$, the squared-lengths of w, x, and $w-x$ are all preserved up to multiplicative factors of $1 \pm \gamma/16$. This then implies that the cosine of the angle between w and x (i.e., the margin of x with respect to w) is preserved up to an additive factor of $\pm\gamma/4$. Specifically, using $\hat{x} = \frac{x}{||x||}$ and $\hat{w} = \frac{w}{||w||}$, which implies $\frac{\hat{F}(w)\cdot\hat{F}(x)}{||\hat{F}(w)||\ ||\hat{F}(x)||} = \frac{\hat{F}(\hat{w})\cdot\hat{F}(\hat{x})}{||\hat{F}(\hat{w})||\ ||\hat{F}(\hat{x})||}$, we have:

$$\begin{aligned}\frac{\hat{F}(\hat{w})\cdot\hat{F}(\hat{x})}{||\hat{F}(\hat{w})||\ ||\hat{F}(\hat{x})||} &= \frac{\frac{1}{2}(||\hat{F}(\hat{w})||^2 + ||\hat{F}(\hat{x})||^2 - ||\hat{F}(\hat{w}) - \hat{F}(\hat{x})||^2)}{||\hat{F}(\hat{w})||\ ||\hat{F}(\hat{x})||} \\ &\in [\hat{w}\cdot\hat{x} - \gamma/4, \hat{w}\cdot\hat{x} + \gamma/4].\end{aligned}$$

In other words, we have shown the following:

$$\text{For all } x, \quad \Pr_A\left[\left|\frac{w\cdot x}{||w||\ ||x||} - \frac{\hat{F}(w)\cdot\hat{F}(x)}{||\hat{F}(w)||\ ||\hat{F}(x)||}\right| \geq \gamma/4\right] \leq \varepsilon\delta/4.$$

Since the above is true for all x, it is clearly true for random x from $F_2(D)$. So,

$$\Pr_{x\leftarrow F_2(D),A}\left[\left|\frac{w\cdot x}{||w||\ ||x||} - \frac{\hat{F}(w)\cdot\hat{F}(x)}{||\hat{F}(w)||\ ||\hat{F}(x)||}\right| \geq \gamma/4\right] \leq \varepsilon\delta/4,$$

which implies that:

$$\Pr_A\left[\Pr_{x\leftarrow F_2(D)}\left(\left|\frac{w\cdot x}{||w||\ ||x||} - \frac{\hat{F}(w)\cdot\hat{F}(x)}{||\hat{F}(w)||\ ||\hat{F}(x)||}\right| \geq \gamma/4\right) \geq \varepsilon/2\right] \leq \delta/2.$$

Since w has error at most $\varepsilon/2$ at margin $\gamma/2$, this then implies that the probability that $\hat{F}(w)$ has error more than ε over $\hat{F}(F_2(D))$ at margin $\gamma/4$ is at most $\delta/2$. Combining this with the $\delta/2$ failure probability of F_2 completes the proof. □

As before, the running time to compute our mappings is polynomial in $1/\gamma, 1/\varepsilon, 1/\delta$ and the time to compute the kernel function K.

Since the dimension d_3 of the mapping in Theorem 3 is only logarithmic in $1/\varepsilon$, this means we can set ε to be small enough so that with high probability, a sample of size $O(d_3 \log d_3)$ would be perfectly separable. This means we could use *any* noise-free linear-separator learning algorithm in R^{d_3} to learn the target concept. However, this requires using $d_2 = \tilde{O}(1/\gamma^4)$ (i.e., $\tilde{O}(1/\gamma^4)$ unlabeled examples to construct the mapping).

Corollary 2. *Given $\varepsilon', \delta, \gamma < 1$, if P has margin γ in the ϕ-space, then $\tilde{O}(\frac{1}{\varepsilon'\gamma^4})$ unlabeled examples are sufficient so that with probability $1-\delta$, mapping $F_3 : X \to R^{d_3}$ has the property that $F_3(P)$ is linearly separable with error $o(\varepsilon'/(d_3 \log d_3))$, where $d_3 = O(\frac{1}{\gamma^2} \log \frac{1}{\varepsilon'\gamma\delta})$.*

Proof. Just plug in the desired error rate into the bounds of Theorem 3. □

4.1 A few extensions

So far, we have assumed that the distribution P is perfectly separable with margin γ in the ϕ-space. Suppose, however, that P is only separable with error α at margin γ. That is, there exists a vector w in the ϕ-space that correctly classifies a $1-\alpha$ probability mass of examples by margin at least γ, but the remaining α probability mass may be either within the margin or incorrectly classified. In that case, we can apply all the previous results to the $1-\alpha$ portion of the distribution that is correctly separated by margin γ, and the remaining α probability mass of examples may or may not behave as desired. Thus all preceding results (Lemma 1, Corollary 1, Theorem 1, and Theorem 3) still hold, but with ε replaced by $(1-\alpha)\varepsilon + \alpha$ in the error rate of the resulting mapping.

Another extension is to the case that the target separator does not pass through the origin: that is, it is of the form $w \cdot \phi(x) \geq \beta$ for some value β. If ϕ is normalized, so that $||\phi(x)|| = 1$ for all $x \in X$, then all results carry over directly. In particular, all our results follow from arguments showing that the cosine of the angle between w and $\phi(x)$ changes by at most ε due to the reduction in dimension. If $\phi(x)$ is not normalized, then all results carry over with γ replaced by γ/R, where R is an upper bound on $||\phi(x)||$, as is done with standard margin bounds [3, 21, 11].

5 On the necessity of access to D

Our algorithms construct mappings $F : X \to R^d$ using black-box access to the kernel function $K(x, y)$ together with unlabeled examples from the input distribution D. It is natural to ask whether it might be possible to remove the need for access to D. In particular, notice that the mapping resulting from the Johnson-Lindenstrauss lemma has nothing to do with the input distribution: if we have access to the ϕ-space, then no matter what the distribution is, a random projection down to R^d will approximately preserve the existence of a large-margin separator with high probability.[5] So perhaps such a mapping F can be produced by just computing K on some polynomial number of cleverly-chosen (or uniform random) points in X. (Let us assume X is a "nice" space such as the unit ball or $\{0,1\}^n$ that can be randomly sampled.) In this section, we show this is not possible in general for an arbitrary black-box kernel. This leaves open, however, the case of specific natural kernels.

One way to view the result of this section is as follows. If we define a feature space based on uniform binary (Rademacher) or gaussian-random points in the ϕ-space, then we know this will work by the Johnson-Lindenstrauss lemma. If we define features based on points in $\phi(X)$ (the image of X under ϕ) chosen according to $\phi(D)$, then this will work by Corollary 1. However, if we define features based on points in $\phi(X)$ chosen according to some method that does not depend on D, then there will exist kernels for which this does not work.

In particular, we demonstrate the necessity of access to D as follows. Consider $X = \{0,1\}^n$, let X' be a random subset of $2^{n/2}$ elements of X, and let D be the uniform distribution on X'. For a given target function c, we will define a special ϕ-function ϕ_c such that c is a large margin separator in the ϕ-space under distribution D, but that only the points in X' behave nicely, and points not in X' provide no useful information. Specifically, consider $\phi_c : X \to R^2$ defined as:

$$\phi_c(x) = \begin{cases} (1,0) & \text{if } x \notin X' \\ (-1/2, \sqrt{3}/2) & \text{if } x \in X' \text{ and } c(x) = 1 \\ (-1/2, -\sqrt{3}/2) & \text{if } x \in X' \text{ and } c(x) = -1 \end{cases}$$

See figure 1. This then induces the kernel:

$$K_c(x,y) = \begin{cases} 1 & \text{if } x, y \notin X' \text{ or } [x, y \in X' \text{ and } c(x) = c(y)] \\ -1/2 & \text{otherwise} \end{cases}$$

Notice that the distribution $P = (D, c)$ over labeled examples has margin $\gamma = \sqrt{3}/2$ in the ϕ-space.

Theorem 4. *Suppose an algorithm makes polynomially many calls to a black-box kernel function over input space $\{0,1\}^n$ and produces a mapping $F : X \to R^d$ where d is polynomial in n. Then for random X' and random c in the above construction, with high probability $F(P)$ will not even be weakly-separable (even though P has margin $\gamma = \sqrt{3}/2$ in the ϕ-space).*

[5] To be clear about the order of quantification, the statement is that for any distribution, a random projection will work with high probability. However, for any given projection, there may exist bad distributions. So, even if we could define a mapping of the sort desired, we might still expect the algorithm to be randomized.

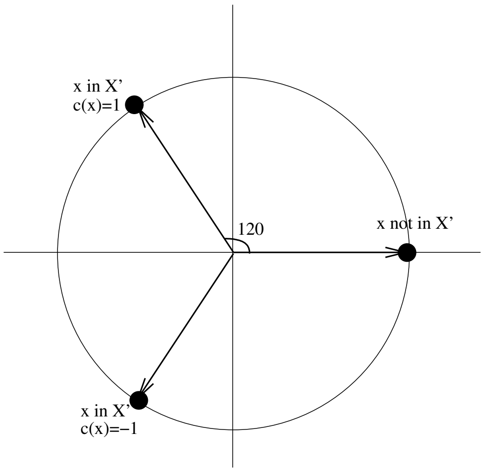

Fig. 1. Function ϕ_c used in lower bound.

Proof. Consider any algorithm with black-box access to K attempting to create a mapping $F : X \to R^d$. Since X' is a random exponentially-small fraction of X, with high probability all calls made to K when constructing the function F are on inputs not in X'. Let us assume this indeed is the case. This implies that (a) all calls made to K when constructing the function F return the value 1, and (b) at "runtime" when x chosen from D (i.e., when F is used to map training data), even though the function $F(x)$ may itself call $K(x, y)$ for different previously-seen points y, these will all give $K(x, y) = -1/2$. In particular, this means that $F(x)$ is independent of the target function c. Finally, since X' has size $2^{n/2}$ and d is only polynomial in n, we have by simply counting the number of possible partitions of $F(X')$ by halfspaces that with high probability $F(P)$ will not even be weakly separable for a random function c over X'. Specifically, for any given halfspace, the probability over choice of c that it has error less than $1/2 - \epsilon$ is exponentially small in $|X'|$ (by Hoeffding bounds), which is doubly-exponentially small in n, whereas there are "only" $2^{O(dn)}$ possible partitions by halfspaces. □

Notice that the kernel in the above argument is positive semidefinite. If we wish to have a positive definite kernel, we can simply change "1" to "$1 - \alpha$" and "$-1/2$" to "$-\frac{1}{2}(1 - \alpha)$" in the definition of $K(x, y)$, except for $y = x$ in which case we keep $K(x, y) = 1$. This corresponds to a function ϕ in which rather that mapping points exactly into R^2, we map into R^{2+2^n} giving each example a $\sqrt{\alpha}$-component in its own dimension, and we scale the first two components by $\sqrt{1 - \alpha}$ to keep $\phi_c(x)$ a unit vector. The margin now becomes $\frac{\sqrt{3}}{2}(1 - \alpha)$. Since the modifications provide no real change (an algorithm with access to the original kernel can simulate this one), the above arguments apply to this kernel as well.

One might complain that the kernels used in the above argument are not efficiently computable. However, this can be rectified (assuming the existence of one-way functions) by defining X' to be a cryptographically pseudorandom subset of X and c to be a pseudorandom function [13]. In this case, except for the very last step, the above argument still holds for polynomial-time algorithms. The only issue, which arises in the last step, is that we do not know any polynomial-time algorithm to test if $F(P)$ is weakly-separable in R^d (which would distinguish c from a truly-random function and provide the needed contradiction). Thus, we would need to change the conclusion of the theorem to be that "$F(P)$ is not even *weakly-learnable* by a polynomial time algorithm".

Of course, these kernels are extremely unnatural, each with its own hidden target function built in. It seems quite conceivable that positive results independent of the distribution D can be achieved for standard, natural kernels.

6 Experiments

One consequence of our analysis is that it provides an alternative to "kernelizing" a learning algorithm: rather than modifying the algorithm to use kernels, one can instead construct a mapping into a low-dimensional space using the kernel and the data distribution, and then run an un-kernelized algorithm over examples in the new space.

To illustrate this idea, we performed several experiments on both synthetic and standard datasets using standard kernel functions. For each experiment, we used unlabeled examples to determine new representations of the data via the mappings F_1 and F_2 described in Section 3. Then, to find linear decision surfaces in these new feature spaces (and so to come up with classification rules for our learning problem) we used both the Balanced Winnow algorithm (see [17], [19]), as well as linear SVM. We compared the accuracies of these methods with those produced by SVM with the same kernel K. We used the SVM implementation available at [27] and described in [12].[6]

Synthetic datasets

To test our methods, we generated several synthetic datasets as follows. We started by considering 2-dimensional input data with separating boundaries of the form $\frac{x_1^2}{a^2} + \frac{x_2^2}{b^2} - 1 = 0$ or $\frac{x_1^2}{a^2} - \frac{x_2^2}{b^2} - 1 = 0$. We generated points according to various distributions on which we ensured that there is a reasonable margin in the ϕ-space induced by the degree-2 polynomial kernel. Specifically, we generated points $x = (x_1, x_2)$ so that they satisfy $l(x)\left(\frac{x_1^2}{a^2} \pm \frac{x_2^2}{b^2} - 1\right) \geq c$, for various

[6] In all the experiments we report we considered $C = 10$; notice that if the kernel is ideal in the sense that the data is perfectly separable in the ϕ-space, then $C = \infty$ is the right choice for running SVM under that kernel. However, we cannot expect the data to be perfectly linearly separable in the new feature spaces and, therefore, for running linear SVM under mappings F_1 and F_2 it makes sense to lower the value of the parameter C.

Table 1. Classification errors of the five methods using the second degree polynomial kernel on various synthetic datasets.

TYPE SURFACE	a	b	c	M	d_1	N_{train}	N_{test}	F_1 WINNOW	F_1 SVM	F_2 WINNOW	F_2 SVM	SVM
ELLIPSIS UNIF-UNIF	1	1	0.2	1.2	10	40	100	0.044	0.014	0.031	0.018	0.017
ELLIPSIS GAUSS-UNIF	1	1	0.2	1.2	10	40	100	0.049	0.021	0.013	0.000	0.001
HYPERBOLA UNIF-UNIF	1	1	0.2	1.2	10	40	100	0.031	0.022	0.023	0.012	0.010
HYPERBOLA GAUSS-UNIF	1	1	0.2	1.2	10	40	100	0.004	0.000	0.001	0.000	0.000
ELLIPSIS UNIF-UNIF	1	0.5	0.1	1.1	10	40	100	0.140	0.060	0.065	0.051	0.045
ELLIPSIS GAUSS-UNIF	1	0.5	0.1	1.1	10	40	100	0.061	0.048	0.049	0.032	0.036
HYPERBOLA UNIF-UNIF	1	0.5	0.1	1.1	10	40	100	0.029	0.035	0.018	0.027	0.018
HYPERBOLA GAUSS-UNIF	1	0.5	0.1	1.1	10	40	100	0.008	0.000	0.004	0.000	0.000

Table 2. Classification errors of the five methods on various standard datasets.

DATASET	KERNEL	SIZE(DATA)	d_1	N_{train}	N_{test}	F_1 WINNOW	F_1 SVM	F_2 WINNOW	F_2 SVM	SVM
CANCER	POLY2	683	20	200	513	0.1037	0.0842	0.0713	0.0821	0.0713
IONOSPHERE	POLY2	351	20	250	81	0.1500	0.1160	0.1457	0.1179	0.1278
IRIS 1VS23	POLY1	150	10	50	90	0.0656	0.0144	0.0011	0.0000	0.000
IRIS 2VS13	RBF, $\sigma = 1$	150	10	50	90	0.0767	0.0611	0.0678	0.0444	0.0456
IRIS 3VS12	RBF, $\sigma = 1$	150	10	50	90	0.0733	0.0622	0.0678	0.0556	0.0533

parameters $a, b, c \geq 0$ and we also constrained that $|x_i| \leq M$. This in turn implied that the margin γ in the ϕ-space is at least $\gamma_l = \frac{c}{(1+2M^2)\cdot\sqrt{1+1/a^4+1/b^4}}$. We then picked d_1 random unlabeled examples to define our mappings, and N_{train} random labeled training points to train the classifiers. We ran experiments in this setting for several values of a, b, c, M, using either the uniform distribution inside the legal regions, or a (truncated) gaussian (with different standard deviation parameters). We summarize in Table 1 a few such results for several values of the parameters.[7] For all five methods (mapping F_1 with Winnow, mapping F_1 with linear SVM, mapping F_2 with Winnow, mapping F_2 with linear SVM, and SVM) we report the average errors on a random test set over 10 runs of the experiment. Note that the choices of d_1 and N_{train} for the experiments we summarize in Table 1 are substantially smaller than those given by the theoretical bounds, but performance appears to still be quite reasonable especially under mapping F_2.

[7] To be more explicit, for the experiments we report in Table 1, we consider a 50/50 distribution. To generate a random point $x = (x_1, x_2)$ for the ellipsis unif-unif case we first flip an unbiased coin to decide its sign $l(x)$, and then pick a point uniformly at random in the region specified by $l(x)(\frac{x_1^2}{a^2} + \frac{x_2^2}{b^2} - 1) \geq c$, $|x_i| \leq M$. For the ellipsis gauss-unif case we similarly first flip an unbiased coin to decide the sign $l(x)$, and then if $l(x)$ is 1 we pick a point uniformly at random in the region specified by $(\frac{x_1^2}{a^2} + \frac{x_2^2}{b^2} - 1) \geq c$, $|x_i| \leq M$; if $l(x)$ is -1 we keep generating points $x = (x_1, x_2)$ with x_i distributed gaussian with mean 0 and variance 0.2 until we have $(\frac{x_1^2}{a^2} + \frac{x_2^2}{b^2} - 1) \leq -c$, $|x_i| \leq M$. In a similar way we obtain random points for the hyperbola unif-unif and hyperbola gauss-unif cases.

Standard datasets

We also compared our mappings with SVM on standard datasets from the UCI Irvine Machine Learning Repository [7], namely Cancer[8], Ionosphere, and IRIS dataset. Both Cancer and Ionosphere datasets are for binary classification problems. IRIS is a dataset with three classes, Iris Setosa, Iris Versicolor and Iris Virginica, and as in [12], we constructed three binary classification problems associated with it: separating Setosa from the other classes, which we call IRIS 1VS23, separating Versicolor from the other classes, which we call IRIS 2VS13, and separating Virginica from the other classes, which we call IRIS 3VS12.

In Table 2 we summarize several results obtained as follows. For each dataset, we first randomly permute all its examples, we then pick d_1 unlabeled points for creating our mappings, and then from the remaining we pick N_{train} examples for training and keep the rest for testing. We repeat the procedure 10 times and then report for all five methods the average error on the test set. We use a polynomial kernel of degree 2 for Cancer and Ionosphere datasets, an RBF kernel with $\sigma = 1$ for IRIS 2VS13 and for IRIS 3VS12, and a linear kernel for IRIS 1VS23 (for IRIS dataset we considered kernels suggested in [12]).

Notice that in most of the cases both Winnow and linear SVM performed nearly as well in the new feature spaces. An interesting point to observe is that mapping F_2 performs nearly as well as SVM, while on several datasets mapping F_1 performs slightly worse. This is to some extent expected since under mapping F_1 we do not expect to have large margin, and also the size of our training set is usually quite small.

Discussion

The experiments show that (at least for this data) mappings F_1 and F_2 can be used to place data into a low-dimensional space and run a linear-separator algorithm (Winnow or linear SVM) without much degradation in performance. Note that we did not experience any *improvement* in performance. However, the ability to perform such explicit mappings opens the door to other possible learning algorithms, perhaps especially designed for low-dimensional data or especially designed for speed, that one might not be able to run over the original data representation. In particular, these mappings allow one to enjoy the benefits of having a large margin in the ϕ-space without restricting the class of learning algorithms to those that are easily kernelizable.

7 Conclusions and Open Problems

We show how given black-box access to a kernel function K and a distribution D (i.e., unlabeled examples) we can use K and D together to *efficiently* construct a new low-dimensional feature space in which to place the data that approximately preserves the desired properties of the kernel. Our procedure uses two types of

[8] Note that we discarded from this dataset those examples with missing attributes.

"random" mappings. The first is a mapping based on random examples drawn from D that is used to construct the intermediate space, and the second is a mapping based on Rademacher/binary (or Gaussian) random vectors in the intermediate space as in the Johnson-Lindenstrauss lemma.

Our analysis suggests that designing a good kernel function is much like designing a good feature space. It also provides an alternative to "kernelizing" a learning algorithm: rather than modifying the algorithm to use kernels, one can instead construct a mapping into a low-dimensional space using the kernel and the data distribution, and then run an un-kernelized algorithm over examples drawn from the mapped distribution.

One interesting aspect of our simplest method, namely choosing $x_1, \ldots, x_d$ from D and then using the mapping $x \mapsto (K(x, x_1), \ldots, K(x, x_d))$, is that it can be applied to any generic "similarity" function $K(x, y)$, even those that are not necessarily legal kernels and do not necessarily have the same interpretation as computing a dot-product in some implicit ϕ-space. It would be interesting if one could prove guarantees for this more general setting.

Our main concrete open question is whether, for natural standard kernel functions, one can produce mappings $F : X \to R^d$ in an oblivious manner, without using examples from the data distribution. The Johnson-Lindenstrauss lemma tells us that such mappings exist, but the goal is to produce them without explicitly computing the ϕ-function. Barring that, perhaps one can at least reduce the unlabeled sample-complexity of our approach.

On the practical side, it would be interesting to further explore the alternatives that these (or other) mappings provide to widely used algorithms such as SVM, or Kernel Perceptron.

Acknowledgements

We would like to thank Adam Kalai and John Langford for helpful discussions. We would also like to thank the anonymous referees for their many useful comments and suggestions which helped to improve the presentation of our results. This work was supported in part by NSF grants CCR-0105488, NSF-ITR CCR-0122581, and NSF-ITR IIS-0312814.

References

1. D. Achlioptas, "Database-friendly Random Projections", Journal of Computer and System Sciences, Volume 66, Issue 4, pp. 671–687, 2003.
2. R. I. Arriaga, S. Vempala, "An Algorithmic Theory of Learning, Robust Concepts and Random Projection", Proceedings of the 40th Foundations of Computer Science, pp. 616–623, 1999. Journal version to appear in Machine Learning.
3. P. Bartlett, J. Shawe-Taylor, "Generalization Performance of Support Vector Machines and Other Pattern Classifiers", Advances in Kernel Methods: Support Vector Learning, pp. 43–54, MIT Press, 1999.
4. S. Ben-David, N. Eiron, H.U. Simon, "Limitations of Learning Via Embeddings in Euclidean Half-Spaces", Journal of Machine Learning Research, Volume 3, pp. 441–461, 2003.

5. S. Ben-David, “A Priori Generalization Bounds for Kernel Based Learning”, NIPS 2001 Workshop on Kernel Based Learning.
6. A. Ben-Israel, T.N.E. Greville, “Generalized Inverses: Theory and Applications”, Wiley, New York, 1974.
7. C.L. Blake and C. J. Merz, “UCI Repository of Machine Learning Databases”. [http://www.ics.uci.edu/ mlearn/MLRepository.html], 1998.
8. B. E. Boser, I. M. Guyon, V. N. Vapnik, “A Training Algorithm for Optimal Margin Classifiers”, Proceedings of the Fifth Annual Workshop on Computational Learning Theory, pp. 144–152, 1992.
9. C. Cortes, V. Vapnik, “Support-Vector Networks”, Machine Learning, Volume 20, No. 3, pp. 273–297, 1995.
10. S. Dasgupta, A. Gupta, “An Elementary Proof of the Johnson-Lindenstrauss Lemma”, Tech Report, UC Berkeley, 1999.
11. Y. Freund, R. E. Schapire, “Large Margin Classification Using the Perceptron Algorithm”, Machine Learning, Volume 37, No. 3, pp. 277–296, 1999.
12. S.R. Gunn, “Support Vector Machines for Classification and Regression”, Technical Report, Image Speech and Intelligent Systems Research Group, University of Southampton, 1997.
13. O. Goldreich, S. Goldwasser, S. Micali, “How to Construct Random Functions”, Journal of the ACM, Volume 33, No. 4,pp. 792–807, 1986.
14. P. Indyk and R. Motwani, “Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality,” Proceedings of the 30th Annual ACM Symposium on Theory of Computing, pp. 604–613, 1998.
15. R. Herbrich, “Learning Kernel Classifiers”, MIT Press, Cambridge, 2002.
16. W. B. Johnson, J. Lindenstrauss, “ Extensions of Lipschitz Mappings into a Hilbert Space”, Contemporary Mathematics, Volume 26, pp. 189–206, 1984.
17. N. Littlestone. “Learning Quickly when Irrelevant Attributes Abound: A New Linear-threshold Algorithm”. Machine Learning, Volume 2, Issue 4, pp. 285–318, 1988.
18. K. R. Muller, S. Mika, G. Ratsch, K. Tsuda, B. Scholkopf, “An Introduction to Kernel-based Learning Algorithms”, IEEE Transactions on Neural Networks, Volume 12, Issue 2, pp. 181–201, 2001.
19. Z. Nevo, R. El-Yaniv, “On Online Learning of Decision Lists”, The Journal of Machine Learning Research, Volume 3, pp. 271–301, 2003.
20. B. Scholkopf, C. J. C. Burges, S. Mika, “Advances in Kernel Methods: Support Vector Learning”, MIT Press, 1999.
21. J. Shawe-Taylor, P. L. Bartlett, R. C. Williamson, M. Anthony, “Structural Risk Minimization over Data-Dependent Hierarchies”, IEEE Transactions on Information Theory, Volume 44(5), pp. 1926–1940, 1998.
22. J. Shawe-Taylor, N. Cristianini, “Kernel Methods for Pattern Analysis”, Cambridge University Press, 2004.
23. B. Scholkopf, K. Tsuda, J.-P. Vert, “Kernel Methods in Computational Biology”, MIT Press, 2004.
24. A. J. Smola, P. Bartlett, B. Scholkopf, D. Schuurmans (Eds.), “Advances in Large Margin Classifiers”, MIT Press, 2000.
25. B. Scholkopf, A. J. Smola , “Learning with Kernels. Support Vector Machines, Regularization, Optimization, and Beyond”, MIT University Press, Cambridge, 2002.
26. V. N. Vapnik, “Statistical Learning Theory”, John Wiley and Sons Inc., New York, 1998.
27. http://www.isis.ecs.soton.ac.uk/resources/svminfo/