Random Projection, Margins, Kernels, and Feature-Selection

Avrim Blum

Department of Computer Science
Carnegie Mellon University
Pittsburgh PA 15213-3891

Abstract. Random projection is a simple technique that has had a number of applications in algorithm design. In the context of machine learning, it can provide insight into questions such as "why is a learning problem easier if data is separable by a large margin?" and "in what sense is choosing a kernel much like choosing a set of features?" This talk is intended to provide an introduction to random projection and to survey some simple learning algorithms and other applications to learning based on it. I will also discuss how, given a kernel as a black-box function, we can use various forms of random projection to extract an explicit small feature space that captures much of what the kernel is doing. This talk is based in large part on work in [BB05,BBV04] joint with Nina Balcan and Santosh Vempala.

1 Introduction

Random projection is a technique that has found substantial use in the area of algorithm design (especially *approximation algorithms*), by allowing one to substantially reduce dimensionality of a problem while still retaining a significant degree of problem structure. In particular, given n points in Euclidean space (of any dimension but which we can think of as R^n), we can project these points down to a random d-dimensional subspace for $d \ll n$, with the following outcomes:

1. If $d = \omega(\frac{1}{\gamma^2} \log n)$ then Johnson-Lindenstrauss type results (described below) imply that with high probability, relative distances and angles between all pairs of points are approximately preserved up to $1 \pm \gamma$.
2. If $d = 1$ (i.e., we project points onto a random line) we can often still get something useful.

Projections of the first type have had a number of uses including fast approximate nearest-neighbor algorithms [IM98,EK00] and approximate clustering algorithms [Sch00] among others. Projections of the second type are often used for "rounding" a semidefinite-programming relaxation, such as for the Max-CUT problem [GW95], and have been used for various graph-layout problems [Vem98].

The purpose of this survey is to describe some ways that this technique can be used (either practically, or for providing insight) in the context of machine learning. In particular, random projection can provide a simple way to see why data that is separable by a large *margin* is easy for learning even if data lies in a high-dimensional space (e.g., because such data can be randomly projected down to a low dimensional space without affecting separability, and therefore it is "really" a low-dimensional problem after all). It can also suggest some especially simple algorithms. In addition, random projection (of various types) can be used to provide an interesting perspective on *kernel* functions, and also provide a mechanism for converting a kernel function into an explicit feature space.

The use of Johnson-Lindenstrauss type results in the context of learning was first proposed by Arriaga and Vempala [AV99], and a number of uses of random projection in learning are discussed in [Vem04]. Experimental work on using random projection has been performed in [FM03,GBN05,Das00]. This survey, in addition to background material, focuses primarily on work in [BB05,BBV04]. Except in a few places (e.g., Theorem 1, Lemma 1) we give only sketches and basic intuition for proofs, leaving the full proofs to the papers cited.

1.1 The setting

We are considering the standard PAC-style setting of supervised learning from i.i.d. data. Specifically, we assume that examples are given to us according to some probability distribution D over an instance space X and labeled by some unknown target function $c : X \to \{-1, +1\}$. We use $P = (D, c)$ to denote the combined distribution over labeled examples. Given some sample S of labeled training examples (each drawn independently from D and labeled by c), our objective is to come up with a hypothesis h with low true error: that is, we want $\Pr_{x \sim D}(h(x) \neq c(x))$ to be low. In the discussion below, by a "learning problem" we mean a distribution $P = (D, c)$ over labeled examples.

In the first part of this survey (Sections 2 and 3), we will think of the input space X as Euclidean space, like R^n. In the second part (Section 4), we will discuss kernel functions, in which case one should think of X as just some abstract space, and a kernel function $K : X \times X \to [-1, 1]$ is then some function that provides a measure of similarity between two input points. Formally, one requires for a legal kernel K that there exist some implicit function ϕ mapping X into a (possibly very high-dimensional) space, such that $K(x, y) = \phi(x) \cdot \phi(y)$. In fact, one interesting property of some of the results we discuss is that they make sense to apply even if K is just an arbitrary similarity function, and not a "legal" kernel, though the theorems make sense only if such a ϕ exists. Extensions of this framework to more general similarity functions are given in [BB06].

Definition 1. *We say that a set S of labeled examples is* **linearly separable by margin** γ *if there exists a unit-length vector w such that:*

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

That is, the separator $w \cdot x \geq 0$ has margin γ if every labeled example in S is correctly classified and furthermore the cosine of the angle between w and x has magnitude at least γ.[1] For simplicity, we are only considering separators that pass through the origin, though results we discuss can be adapted to the general case as well.

We can similarly talk in terms of the distribution P rather than a sample S.

Definition 2. *We say that* P **is linearly separable by margin** γ *if there exists a unit-length vector* w *such that:*

$$\Pr_{(x,\ell)\sim P}[\ell(w \cdot x)/||x|| < \gamma] = 0,$$

and we say that P **is separable with error** α **at margin** γ *if there exists a unit-length vector* w *such that:*

$$\Pr_{(x,\ell)\sim P}[\ell(w \cdot x)/||x|| < \gamma] \leq \alpha.$$

A powerful theoretical result in machine learning is that if a learning problem is linearly separable by a large margin γ, then that makes the problem "easy" in the sense that to achieve good generalization one needs only a number of examples that depends (polynomially) on $1/\gamma$, with no dependence on the dimension of the ambient space X that examples lie in. In fact, two results of this form are:

1. The classic Perceptron Convergence Theorem that the Perceptron Algorithm makes at most $1/\gamma^2$ mistakes on any sequence of examples separable by margin γ [Blo62,Nov62,MP69]. Thus, if the Perceptron algorithm is run on a sample of size $1/(\epsilon\gamma^2)$, the expected error rate of its hypothesis at a random point in time is at most ϵ. (For further results of this form, see [Lit89,FS99]).
2. The more recent margin bounds of [STBWA98,BST99] that state that $|S| = O(\frac{1}{\varepsilon}[\frac{1}{\gamma^2}\log^2(\frac{1}{\gamma\varepsilon})+\log\frac{1}{\delta}])$ is sufficient so that with high probability, *any* linear separator of S with margin γ has true error at most ϵ. Thus, this provides a sample complexity bound that applies to any algorithm that finds large-margin separators.

In the next two sections, we give two more ways of seeing why having a large margin makes a learning problem easy, both based on the idea of random projection.

2 An extremely simple learning algorithm

In this section, we show how the idea of random projection can be used to get an extremely simple algorithm (almost embarrassingly so) for *weak*-learning,

[1] 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 $||x||$ 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.

with error rate $1/2 - \gamma/4$, whenever a learning problem is linearly separable by some margin γ. This can then be plugged into Boosting [Sch90,FS97], to achieve strong-learning. This material is taken from [BB05].

In particular, the algorithm is as follows.

Algorithm 1 (Weak-learning a linear separator)

1. *Pick a random linear separator. Specifically, choose a random unit-length vector h and consider the separator $h \cdot x \geq 0$. (Equivalently, project data to the 1-dimensional line spanned by h and consider labeling positive numbers as positive and negative numbers as negative.)*
2. *Evaluate the error of the separator selected in Step 1. If the error is at most $1/2 - \gamma/4$ then halt with success, else go back to 1.*

Theorem 1. *If P is separable by margin γ then a* random *linear separator will have error at most $\frac{1}{2} - \gamma/4$ with probability $\Omega(\gamma)$.*

In particular, Theorem 1 implies that the above algorithm will in expectation repeat only $O(1/\gamma)$ times before halting.[2]

Proof. Consider a (positive) example x such that $x \cdot w^*/||x|| \geq \gamma$. The angle between x and w^* is then some value $\alpha \leq \pi/2 - \gamma$. Now, a random vector h, when projected onto the 2-dimensional plane defined by x and w^*, looks like a random vector in this plane. Therefore, we have (see Figure 1):

$$\Pr_h(h \cdot x \leq 0 | h \cdot w^* \geq 0) = \alpha/\pi \leq 1/2 - \gamma/\pi.$$

Similarly, for a negative example x, for which $x \cdot w^*/||x|| \leq -\gamma$, we have:

$$\Pr_h(h \cdot x \geq 0 | h \cdot w^* \geq 0) \leq 1/2 - \gamma/\pi.$$

Therefore, if we define $h(x)$ to be the classifier defined by $h \cdot x \geq 0$, we have:

$$\mathbf{E}[err(h) | h \cdot w^* \geq 0] \leq 1/2 - \gamma/\pi.$$

Finally, since the error rate of any hypothesis is bounded between 0 and 1, and a random vector h has a 1/2 chance of satisfying $h \cdot w^* \geq 0$, it must be the case that:

$$\Pr_h[err(h) \leq 1/2 - \gamma/4] = \Omega(\gamma).$$

$\square$

[2] For simplicity, we have presented Algorithm 1 as if it can exactly evaluate the true error of its chosen hypothesis in Step 2. Technically, we should change Step 2 to talk about empirical error, using an intermediate value such as $1/2 - \gamma/6$. In that case, sample size $O(\frac{1}{\gamma^2} \log \frac{1}{\gamma})$ is sufficient to be able to run Algorithm 1 for $O(1/\gamma)$ repetitions, and evaluate the error rate of each hypothesis produced to sufficient precision.

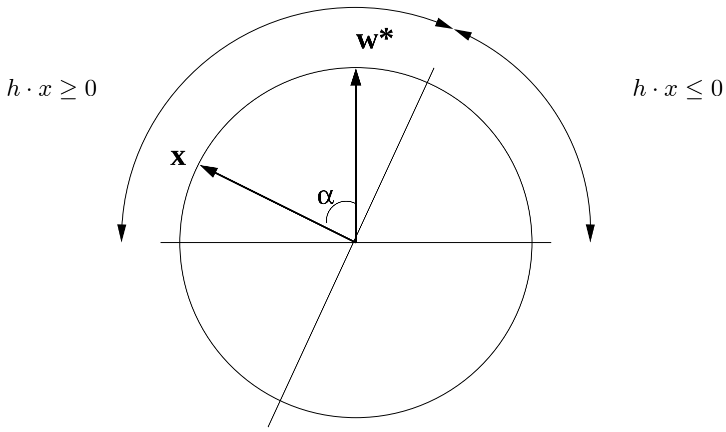

Fig. 1. For random h, conditioned on $h \cdot w^* \geq 0$, we have $\Pr(h \cdot x \leq 0) = \alpha/\pi$ and $\Pr(h \cdot x \geq 0) = 1 - \alpha/\pi$.

3 The Johnson-Lindenstrauss Lemma

The Johnson-Lindenstrauss Lemma [JL84,DG02,IM98] states that given a set S of points in R^n, if we perform an orthogonal projection of those points onto a random d-dimensional subspace, then $d = O(\frac{1}{\gamma^2} \log |S|)$ is sufficient so that with high probability all pairwise distances are preserved up to $1 \pm \gamma$ (up to scaling). Conceptually, one can think of a random projection as first applying a random rotation to R^n and then reading off the first d coordinates. In fact, a number of different forms of "random projection" are known to work (including some that are especially efficient to perform computationally, considered in [Ach03,AV99]). In particular, if we think of performing the projection via multiplying all points, viewed as row-vectors of length n, by an $n \times d$ matrix A, then several methods for selecting A that provide the desired result are:

1. Choosing its columns to be d random orthogonal unit-length vectors (a true random orthogonal projection).
2. Choosing each entry in A independently from a standard Gaussian (so the projection can be viewed as selecting d vectors $u_1, u_2, \ldots, u_d$ from a spherical gaussian and mapping a point p to $(p \cdot u_1, \ldots, p \cdot u_d)$.
3. Choosing each entry in A to be 1 or -1 independently at random.

Some especially nice proofs for the Johnson-Lindenstrauss Lemma are given by Indyk and Motwani [IM98] and Dasgupta and Gupta [DG02]. Here, we just give the basic structure and intuition for the argument. In particular, consider two points p_i and p_j in the input and their difference $v_{ij} = p_i - p_j$. So, we are interested in the length of $v_{ij}A$. Fixing v_{ij}, let us think of each of the d coordinates $y_1, \ldots, y_d$ in the vector $y = v_{ij}A$ as random variables (over the choice of A). Then, in each form of projection above, these d random variables are nearly independent (in fact, in forms (2) and (3) they are completely independent random variables). This allows us to use a Chernoff-style bound to argue that $d = O(\frac{1}{\gamma^2}\log\frac{1}{\delta})$ is sufficient so that with probability $1-\delta$, $y_1^2 + \ldots + y_d^2$ will be within $1 \pm \gamma$ of its expectation. This in turn implies that the length of y is within $1 \pm \gamma$ of its expectation. Finally, using $\delta = o(1/n^2)$ we have by the union bound that with high probability this is satisfied simultaneously for all pairs of points p_i, p_j in the input.

Formally, here is a convenient form of the Johnson-Lindenstrauss Lemma given in [AV99]. Let $N(0,1)$ denote the standard Normal distribution with mean 0 and variance 1, and $U(-1,1)$ denote the distribution that has probability 1/2 on -1 and probability 1/2 on 1.

Theorem 2 (Neuronal RP [AV99]). *Let $u, v \in R^n$. Let $u' = \frac{1}{\sqrt{d}}uA$ and $v' = \frac{1}{\sqrt{d}}vA$ where A is a $n \times d$ random matrix whose entries are chosen independently from either $N(0,1)$ or $U(-1,1)$. Then,*

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

Theorem 2 suggests a natural learning algorithm: first randomly project data into a lower dimensional space, and then run some other algorithm in that space, taking advantage of the speedup produced by working over fewer dimensions. Theoretical results for some algorithms of this form are given in [AV99], and experimental results are given in [FM03,GBN05,Das00].

3.1 The Johnson-Lindenstrauss Lemma and Margins

The Johnson-Lindenstrauss lemma provides a particularly intuitive way to see why one should be able to generalize well from only a small amount of training data when a learning problem is separable by a large margin. In particular, imagine a set S of data in some high-dimensional space, and suppose that we randomly project the data down to R^d. By the Johnson-Lindenstrauss Lemma, $d = O(\gamma^{-2}\log|S|)$ is sufficient so that with high probability, all *angles* between points (viewed as vectors) change by at most $\pm\gamma/2$.[3] In particular, consider projecting all points in S *and* the target vector w^*; if initially data was separable by margin γ, then after projection, since angles with w^* have changed by at most $\gamma/2$, the data is still separable (and in fact separable by margin $\gamma/2$). Thus, this means our problem was in some sense really only a d-dimensional problem after all. Moreover, if we replace the "$\log|S|$" term in the bound for d with "$\log\frac{1}{\epsilon}$", then we can use Theorem 2 to get that with high probability at least a $1-\epsilon$ fraction of S will be separable. Formally, talking in terms of the true distribution P, one can state the following theorem. (Proofs appear in, e.g., [AV99,BBV04].)

[3] The Johnson-Lindenstrauss Lemma talks about relative *distances* being approximately preserved, but it is a straightforward calculation to show that this implies angles must be approximately preserved as well.

Theorem 3. *If P is linearly separable by margin γ, then $d = O\left(\frac{1}{\gamma^2}\log(\frac{1}{\varepsilon\delta})\right)$ is sufficient so that with probability at least $1-\delta$, a random projection down to R^d will be linearly separable with error at most ε at margin $\gamma/2$.*

So, Theorem 3 can be viewed as stating that a learning problem separable by margin γ is really only an "$O(1/\gamma^2)$-dimensional problem" after all.

4 Random projection, Kernel functions, and Feature selection

4.1 Introduction

Kernel functions [BGV92,CV95,FS99,MMR$^+$01,STBWA98,Vap98] have become a powerful tool in Machine Learning. 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, margin bounds imply that if the learning problem has a large margin linear separator in that space, then one can avoid paying a high price in terms of sample size as well.

Combining kernel functions with the Johnson-Lindenstrauss Lemma (in particular, Theorem 3 above), we have that if a learning problem indeed has the large margin property under kernel $K(x, y) = \phi(x) \cdot \phi(y)$, then a *random* linear projection of the "ϕ-space" down to a *low* dimensional space approximately preserves linear separability. This means that for any kernel K under which the learning problem is linearly separable by margin γ in the ϕ-space, 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 now consider is whether, given kernel K as a black-box function, we can in fact produce such a mapping efficiently. The problem with the above observation is that it requires explicitly computing the function $\phi(x)$. Since for a given kernel K, the dimensionality of the ϕ-space might be quite large, this is not efficient.[4] Instead, what we would like is an efficient procedure that given $K(.,.)$ as a black-box program, produces a mapping with the desired properties using 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. This would mean we can effectively convert a kernel K that is good for some learning problem into an explicit set of features, without a need for "kernelizing" our learning algorithm. In this section, we describe several methods for doing so; this work is taken from [BBV04].

Specifically, we will show the following. Given black-box access to a kernel function $K(x, y)$, access to unlabeled examples from distribution D, and parameters γ, ε, and δ, we can in polynomial time construct a mapping $F : X \to R^d$

[4] In addition, it is not totally clear how to apply Theorem 2 if the dimension of the ϕ-space is infinite.

(i.e., to a set of d real-valued features) where $d = O\left(\frac{1}{\gamma^2}\log\frac{1}{\varepsilon\delta}\right)$, such that 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 a margin $\gamma' = \Omega(\gamma)$. (If the original learning problem was separable with error α at margin γ then the induced distribution is separable with error $\alpha + \epsilon$ at margin γ'.)

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 $F(x) = (K(x, x_1), \ldots, K(x, x_d))$.[5] 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. Corollary 1 (in Section 4.3 below) shows 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 these results, a natural question is whether it might be possible to perform mappings of this type without access to the underlying distribution. In Section 4.6 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.

4.2 Additional definitions

In analogy to Definition 2, we will say that P is separable by margin γ in the ϕ-space if there exists a unit-length vector w in the ϕ-space such that $\Pr_{(x,\ell)\sim P}[\ell(w \cdot \phi(x))/||\phi(x)|| < \gamma] = 0$, and similarly that P is separable with error α at margin γ in the ϕ-space if the above holds with "$= 0$" replaced by "$\leq \alpha$".

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 span of these vectors: that is, the set of vectors u that can be written as a linear combination $a_1 v_1 + \ldots + a_k v_k$. Also, for a vector u and a subspace Y, let $\mathrm{proj}(u, Y)$ be the orthogonal projection of u down to Y. So, for instance, $\mathrm{proj}(u, \mathrm{span}(v_1, \ldots, v_k))$ is the orthogonal projection of u 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.

[5] In contrast, the Johnson-Lindenstrauss Lemma as presented in Theorem 2 would draw d Gaussian (or uniform $\{-1, +1\}$) random points $r_1, \ldots, r_d$ in the ϕ-space and define $F(x) = (\phi(x) \cdot r_1, \ldots, \phi(x) \cdot r_d)$.

4.3 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 mapping $F : X \to R^d$ with the following property: if the target function indeed has margin γ in the ϕ-space, then with high probability F approximately preserves linear separability. In this section, we analyze two methods that both produce a space of dimension $O(\frac{1}{\varepsilon}[\frac{1}{\gamma^2} + \ln\frac{1}{\delta}])$, such that with probability $1-\delta$ the result is separable with error at most ε. 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 4.4.

The following lemma is key to our analysis.

Lemma 1. *Consider any distribution over labeled examples in Euclidean space such that there exists a linear separator $w \cdot x = 0$ with margin γ. If we draw*

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

examples $z_1, \ldots, z_d$ iid 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 [STBWA98,BST99] 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 [STBWA98,BST99] 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 span(S), let $z_{out}(S) = z - z_{in}(S)$ be the portion of z orthogonal to span(S), and let $z' = z_{out}(S)/||z_{out}(S)||$. Now, for $S' = S \cup \{z\}$, we have $w_{out}(S') = w_{out}(S) - \text{proj}(w_{out}(S), \text{span}(S')) = w_{out}(S) - (w_{out}(S) \cdot z')z'$, where the last equality holds because $w_{out}(S)$ is orthogonal to span(S) and so its projection onto 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. $\square$

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 $x_1, \ldots, x_n$ 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 $F_1(P)$ on labeled examples in R^d 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 $\frac{8}{\varepsilon}[\frac{1}{\gamma^2} + \ln \frac{1}{\delta}]$ unlabeled examples from D as in Corollary 1. 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 $\text{span}(\phi(x_1), \ldots, \phi(x_d))$. Technically, if U is not full rank then we want to use the (Moore-Penrose) pseudoinverse [BIG74] of U in place of U^{-1}.

By Lemma 1, this mapping F_2 maintains approximate separability at margin $\gamma/2$ (See [BBV04] for a full proof):

Theorem 4. *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 $\gamma/2$.*

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.

4.4 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 an additional Johnson-Lindenstrauss style mapping to reduce dimensionality even further. 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.

In particular, let $F_2 : X \to R^{d_2}$ be the mapping from Section 4.3 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. Then consider the 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. In particular, we have (see [BBV04] for a full proof):

Theorem 5. *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* $\gamma/4$*.*

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 5 is only logarithmic in $1/\varepsilon$, this means that if P is perfectly separable with margin γ in the ϕ-space, 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. That is, we could use an arbitrary 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* $d_3 = O(\frac{1}{\gamma^2}\log\frac{1}{\varepsilon'\gamma\delta})$*.*

4.5 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 4, and Theorem 5) 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 our kernel function is normalized, so that $||\phi(x)|| = 1$ for all $x \in X$, then all results carry over directly (note that one can normalize any kernel K by defining $\hat{K}(x, x') = K(x, x')/\sqrt{K(x, x)K(x', x')}$). 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 the kernel is not normalized, then results still carry over if one is willing to divide by the maximum value of $||\phi(x)||$, but we do not know if results carry over if one wishes to be truly translation-independent, say bounding only the radius of the smallest ball enclosing all $\phi(x)$ but not necessarily centered at the origin.

4.6 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.[6] 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 dot-products with uniform or gaussian-random points in the ϕ-space, then we know this will work by the Johnson-Lindenstrauss lemma. However, this requires explicit access to the ϕ-space. Alternatively, using Corollary 1 we can define features based on dot-products with points $\phi(x)$ for $x \in X$, which only requires *implicit* access to the ϕ-space through the kernel. However, this procedure needs to use D to select the points x. What we show here is that such use of D is necessary: if we define features based on points $\phi(x)$ for uniform random $x \in X$, or any other distribution that does not depend on D, then there will exist kernels for which this does not work.

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 2. 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 6. *Suppose an algorithm makes polynomially many calls to a black-box kernel function over input space $\{0,1\}^n$ and produces a mapping $F : X \to R^d$*

[6] 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 would still expect the algorithm to be randomized.

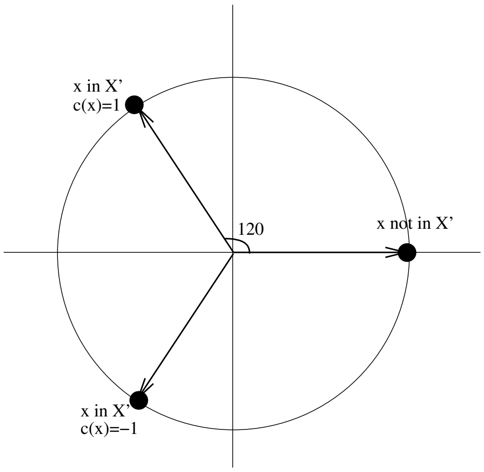

Fig. 2. Function ϕ_c used in lower bound.

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).

Proof Sketch: 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. $\square$

The above construction corresponds to a scenario in which "real data" (the points in X') are so sparse and special that an algorithm without access to D is not able to construct anything that looks even remotely like a real data point by itself (e.g., examples are pixel images of outdoor scenes and yet our poor learning algorithm has no knowledge of vision and can only construct white noise). Furthermore, it relies on a kernel that only does something interesting on the real data (giving nothing useful for $x \notin X'$). It is conceivable that positive results independent of the distribution D can be achieved for standard, natural kernels.

5 Conclusions and Open Problems

This survey has examined ways in which random projection (of various forms) can provide algorithms for, and insight into, problems in machine learning. For example, if a learning problem is separable by a large margin γ, then a random projection to a space of dimension $O(\frac{1}{\gamma^2}\log\frac{1}{\epsilon\delta})$ will with high probability approximately preserve separability, so we can think of the problem as really an $O(1/\gamma^2)$-dimensional problem after all. In addition, we saw that just picking a *random* separator (which can be thought of as projecting to a random 1-dimensional space) has a reasonable chance of producing a weak hypothesis.

We also saw 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 construct a new low-dimensional feature space in which to place the data that approximately preserves the desired properties of the kernel. Thus, through this mapping, we can think of a kernel as in some sense providing a distribution-dependent feature space. One interesting aspect of the simplest method considered, 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. Recent results of [BB06] extend some of these guarantees to this more general setting.

One 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 and Kernel Perceptron.

Acknowledgements

Much of this was based on joint work with Maria-Florina (Nina) Balcan and Santosh Vempala. Thanks also to the referees for their helpful comments. This work was supported in part by NSF grants CCR-0105488, NSF-ITR CCR-0122581, and NSF-ITR IIS-0312814.

References

[Ach03] D. Achlioptas. Database-friendly random projections. *Journal of Computer and System Sciences*, 66(4):671–687, 2003.

[AV99] R. I. Arriaga and S. Vempala. An algorithmic theory of learning, robust concepts and random projection. In *Proceedings of the 40th Annual IEEE Symposium on Foundation of Computer Science*, pages 616–623, 1999.

[BB05] M-F. Balcan and A. Blum. A PAC-style model for learning from labeled and unlabeled data. In *Proceedings of the 18th Annual Conference on Computational Learning Theory (COLT)*, pages 111–126, 2005.

[BB06] M-F. Balcan and A. Blum. On a theory of kernels as similarity functions. Mansucript, 2006.

[BBV04] M.F. Balcan, A. Blum, and S. Vempala. Kernels as features: On kernels, margins, and low-dimensional mappings. In *15th International Conference on Algorithmic Learning Theory (ALT '04)*, pages 194–205, 2004. An extended version is available at `http://www.cs.cmu.edu/~avrim/Papers/`.

[BGV92] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, 1992.

[BIG74] A. Ben-Israel and T.N.E. Greville. *Generalized Inverses: Theory and Applications*. Wiley, New York, 1974.

[Blo62] H.D. Block. The perceptron: A model for brain functioning. *Reviews of Modern Physics*, 34:123–135, 1962. Reprinted in *Neurocomputing*, Anderson and Rosenfeld.

[BST99] P. Bartlett and J. Shawe-Taylor. Generalization performance of support vector machines and other pattern classifiers. In *Advances in Kernel Methods: Support Vector Learning*. MIT Press, 1999.

[CV95] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273 – 297, 1995.

[Das00] S. Dasgupta. Experiments with random projection. In *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 143–151, 2000.

[DG02] S. Dasgupta and A. Gupta. An elementary proof of the Johnson-Lindenstrauss Lemma. *Random Structures & Algorithms*, 22(1):60–65, 2002.

[EK00] Y. Rabani E. Kushilevitz, R. Ostrovsky. Efficient search for approximate nearest neighbor in high dimensional spaces. *SIAM J. Computing*, 30(2):457–474, 2000.

[FM03] D. Fradkin and D. Madigan. Experiments with random projections for machine learning. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 517–522, 2003.

[FS97] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[FS99] Y. Freund and R.E. Schapire. Large margin classification using the Perceptron algorithm. *Machine Learning*, 37(3):277–296, 1999.

[GBN05] N. Goal, G. Bebis, and A. Nefian. Face recognition experiments with random projection. In *Proceedings SPIE Vol. 5779*, pages 426–437, 2005.

[GW95] M.X. Goemans and D.P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, pages 1115–1145, 1995.

[IM98] P. Indyk and R. Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 604–613, 1998.

[JL84] W. B. Johnson and J. Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. In *Conference in Modern Analysis and Probability*, pages 189–206, 1984.

[Lit89] Nick Littlestone. From on-line to batch learning. In *COLT '89: Proceedings of the 2nd Annual Workshop on Computational Learning Theory*, pages 269–284, 1989.

[MMR+01] K. R. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf. An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, 12:181–201, 2001.

[MP69] M. Minsky and S. Papert. *Perceptrons: An Introduction to Computational Geometry*. The MIT Press, 1969.

[Nov62] A.B.J. Novikoff. On convergence proofs on perceptrons. In *Proceedings of the Symposium on the Mathematical Theory of Automata, Vol. XII*, pages 615–622, 1962.

[Sch90] R. E. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.

[Sch00] L. Schulman. Clustering for edge-cost minimization. In *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing*, pages 547–555, 2000.

[STBWA98] J. Shawe-Taylor, P.L. Bartlett, R.C. Williamson, and M. Anthony. Structural risk minimization over data-dependent hierarchies. *IEEE Trans. on Information Theory*, 44(5):1926–1940, 1998.

[Vap98] V. N. Vapnik. *Statistical Learning Theory*. John Wiley and Sons Inc., New York, 1998.

[Vem98] S. Vempala. Random projection: A new approach to VLSI layout. In *Proceedings of the 39th Annual IEEE Symposium on Foundation of Computer Science*, pages 389–395, 1998.

[Vem04] S. Vempala. *The Random Projection Method*. American Mathematical Society, DIMACS: Series in Discrete Mathematics and Theoretical Computer Science, 2004.