On a Theory of Learning with Similarity Functions

Maria-Florina Balcan NINAMF@CS.CMU.EDU
Avrim Blum AVRIM@CS.CMU.EDU
School of Computer Science, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213-3891

Abstract

Kernel functions have become an extremely popular tool in machine learning, with an attractive theory as well. This theory views a kernel as implicitly mapping data points into a possibly very high dimensional space, and describes a kernel function as being good for a given learning problem if data is separable by a large margin in that implicit space. However, while quite elegant, this theory does not directly correspond to one's intuition of a good kernel as a good similarity function. Furthermore, it may be difficult for a domain expert to use the theory to help design an appropriate kernel for the learning task at hand since the implicit mapping may not be easy to calculate. Finally, the requirement of positive semi-definiteness may rule out the most natural pairwise similarity functions for the given problem domain.

In this work we develop an alternative, more general theory of learning with similarity functions (i.e., sufficient conditions for a similarity function to allow one to learn well) that does not require reference to implicit spaces, and does not require the function to be positive semi-definite (or even symmetric). Our results also generalize the standard theory in the sense that any good kernel function under the usual definition can be shown to also be a good similarity function under our definition (though with some loss in the parameters). In this way, we provide the first steps towards a theory of kernels that describes the effectiveness of a given kernel function in terms of natural similarity-based properties.

1. Introduction

Kernel functions have become an extremely popular tool in machine learning, with an attractive theory as well (Shawe-Taylor & Cristianini, 2004; Scholkopf et al., 2004; Herbrich, 2002; Joachims, 2002; Vapnik, 1998). A kernel is a function that takes in two data objects (which could be images, DNA sequences, or points in R^n) and outputs a number, with the property that the function is symmetric and positive-semidefinite. That is, for any kernel $\mathcal{K}$, there must exist an (implicit) mapping ϕ, such that for all inputs x, x' we have $\mathcal{K}(x, x') = \phi(x) \cdot \phi(x')$. The kernel is then used inside a "kernelized" learning algorithm such as SVM or kernel-perceptron as the way in which the algorithm interacts with the data. Typical kernel functions for structured data include the polynomial kernel $\mathcal{K}(x, x') = (1 + x \cdot x')^d$ and the Gaussian kernel $\mathcal{K}(x, x') = e^{-||x-x'||^2/2\sigma^2}$, and a number of special-purpose kernels have been developed for sequence data, image data, and other types of data as well (Cortes & Vapnik, 1995; Cristianini et al., 2001; Lanckriet et al., 2004; Muller et al., 2001; Smola et al., 2000).

The theory behind kernel functions is based on the fact that many standard algorithms for learning linear separators, such as SVMs and the Perceptron algorithm, can be written so that the only way they interact with their data is via computing dot-products on pairs of examples. Thus, by replacing each invocation of $x \cdot x'$ with a kernel computation $\mathcal{K}(x, x')$, the algorithm behaves exactly as if we had explicitly performed the mapping $\phi(x)$, even though ϕ may be a mapping into a very high-dimensional space. Furthermore, these algorithms have convergence rates that depend only on the *margin* of the best separator, and not on the dimension of the space in which the data resides (Anthony & Bartlett, 1999; Shawe-Taylor et al., 1998). Thus, kernel functions are often viewed as providing much of the power of this implicit high-dimensional space, without paying for it computationally (because the ϕ mapping is only implicit) or in terms of sample size (if data is indeed well-separated in that space).

While the above theory is quite elegant, it has a few limitations. First, when designing a kernel function for some learning problem, the intuition typically employed is that a good kernel would be one that serves as a good similarity function for the given problem (Scholkopf et al., 2004). On the other hand, the above theory talks about margins in an implicit and possibly very high-dimensional space. So, in this sense the theory is not that helpful for providing intuition when selecting or designing a kernel function. Second, it may be that the most natural similarity function for a given problem is not positive-semidefinite, and it could require substantial work, possibly reducing the quality of

Appearing in *Proceedings of the 23^{rd} International Conference on Machine Learning*, Pittsburgh, PA, 2006. Copyright 2006 by the author(s)/owner(s).

the function, to coerce it into a legal form. Finally, from a complexity-theoretic perspective, it is a bit unsatisfying for the explanation of the effectiveness of some algorithm to depend on properties of an implicit high-dimensional mapping that one may not even be able to calculate. In particular, the standard theory at first blush has a "something for nothing" feel to it (all the power of the implicit high-dimensional space without having to pay for it) and perhaps there is a more prosaic explanation of what it is that makes a kernel useful for a given learning problem. For these reasons, it would be helpful to have a theory that was in terms of more tangible quantities.

In this paper, we develop a theory of learning with similarity functions that addresses a number of these issues. In particular, we define a notion of what it means for a pairwise function $\mathcal{K}(x, x')$ to be a "good similarity function" for a given learning problem that (a) does not require the notion of an implicit space and allows for functions that are not positive semi-definite, (b) we can show is sufficient to be used for learning, and (c) is broad in that a good kernel in the standard sense (large margin in the implicit ϕ-space) will also satisfy our definition of a good similarity function, though with some loss in the parameters. In this way, we provide the first theory that describes the effectiveness of a given kernel (or more general similarity function) in terms of natural similarity-based properties.

1.1. Our Results and Structure of the Paper

Our main result is a theory for what it means for a pairwise function to be a "good similarity function" for a given learning problem, along with results showing that our main definition is sufficient to be able to learn well and that it captures the standard notion of a good kernel. We begin with a definition (Definition 2) that is especially intuitive and allows for learning via a very simple algorithm, but is fairly restrictive and does not include all kernel functions that induce large-margin separators. We then extend this notion to our main definition (Definition 3) that is somewhat less intuitive, but is now able to capture all functions satisfying the usual notion of a good kernel function and still have implications to learning. Specifically, we show that if $\mathcal{K}$ is a similarity function satisfying Definition 3 then one can algorithmically perform a simple, *explicit* transformation of the data under which there is a low-error large-margin separator. In particular, this transformation involves performing what might be called an "empirical similarity map": selecting a subset of data points as landmarks, and then re-representing the data set based on the similarity of each example to those landmarks. We also consider some variations on this definition that produce somewhat better guarantees on the quality of the final hypothesis produced when combined with known efficient learning algorithms. Finally, in Section 5.1, we describe relationships between our framework and the notion of kernel-target alignment.

A similarity function $\mathcal{K}$ satisfying our definitions is not necessarily guaranteed to produce a good hypothesis when *directly* plugged into standard learning algorithms like SVM or Perceptron (which would be the case if $\mathcal{K}$ satisfied the standard notion of being a good kernel function). Instead, what we show is that such a similarity function can be employed in a 2-stage algorithm: first using $\mathcal{K}$ to re-represent the data as described above, and *then* running a standard (non-kernelized) linear separator algorithm in the new space. One advantage of this, however, is that it allows for the use of a broader class of learning algorithms since one does not need the algorithm used in the second step to be "kernelizable". In fact, this work is motivated by results of (Balcan et al., 2004) that showed how such 2-stage algorithms could be applied as an alternative to kernelizing the learning algorithm in the case of kernel functions.

2. Background and Notation

We consider a learning problem specified as follows. We are given access to labeled examples (x, ℓ) drawn from some distribution P over $X \times \{-1, 1\}$, where X is an abstract instance space. The objective of a learning algorithm is to produce a classification function $g : X \to \{-1, 1\}$ whose error rate $\Pr_{(x,\ell)\sim P}[g(x) \neq \ell]$ is low. We will be considering learning algorithms whose only access to their data is via a pairwise similarity function $\mathcal{K}(x, x')$ that given two examples outputs a number in the range $[-1, 1]$. Specifically,

Definition 1 *A* similarity function *over X is any pairwise function $\mathcal{K} : X \times X \to [-1, 1]$. We say that $\mathcal{K}$ is a symmetric similarity function if $\mathcal{K}(x, x') = \mathcal{K}(x', x)$ for all x, x'.*

Our goal is to give definitions for what it means for a similarity function $\mathcal{K}$ to be "good" for a learning problem P that (ideally) are intuitive, broad, and have the property that a good similarity function results in the ability to learn well. Note that as with the theory of kernel functions, the notion of "goodness" is with respect to a given learning problem P, and *not* with respect to a class of target functions as in the PAC framework.

A similarity function $\mathcal{K}$ is a kernel if there exists a function ϕ from the instance space X into a (possibly implicit) "ϕ-space" such that $\mathcal{K}(x, x') = \phi(x) \cdot \phi(x')$. We say that $\mathcal{K}$ is an (ϵ, γ)-*good kernel function* for a given learning problem P if there exists a vector w in the ϕ-space that has error ϵ at margin γ, where we use a normalized notion of margin, and for simplicity we consider only separators through the origin. Specifically, $\mathcal{K}$ is an (ϵ, γ)-*good kernel function* if there exists a vector w such that

$$\Pr_{(x,\ell(x))\sim P}\left[\ell(x)\frac{\phi(x) \cdot w}{||\phi(x)||\ ||w||} \geq \gamma\right] \geq 1 - \epsilon.$$

We say that $\mathcal{K}$ is a γ-*good kernel function* if it is (ϵ, γ)-good for $\epsilon = 0$; i.e., it has zero error at margin γ. Moreover, we say that $\mathcal{K}$ is a normalized kernel if $\mathcal{K}(x, x) = 1$ for all x.

For simplicity we will assume all kernels are normalized: note that any kernel function $\mathcal{K}$ can be converted to a normalized one $\hat{\mathcal{K}}(x, x') = \frac{\mathcal{K}(x,x')}{\sqrt{\mathcal{K}(x,x)\mathcal{K}(x',x')}}$ without changing its margin properties.

If the standard linear kernel $\mathcal{K}$ defined as $\mathcal{K} = x \cdot x'$ is an (ϵ, γ)-*good kernel function* for a given learning problem P, we say that the learning problem is (ϵ, γ)-*linearly separable*. Moreover, if $||x|| = 1$ for all x we say that P is a normalized (ϵ, γ)-linearly separable problem.

Note that a similarity function need not be a legal kernel. For example, suppose we say two documents have similarity 1 if they have either an author in common or a keyword in common, and otherwise they have similarity 0. Then you could have three documents A, B, and C, such that $\mathcal{K}(A, B) = 1$ because A and B have an author in common, $\mathcal{K}(B, C) = 1$ because B and C have a keyword in common, but $\mathcal{K}(A, C) = 0$ because A and C have neither an author nor a keyword in common (and $\mathcal{K}(A, A) = \mathcal{K}(B, B) = \mathcal{K}(C, C) = 1$). On the other hand, a kernel requires that if $\phi(A)$ and $\phi(B)$ are of unit length and $\phi(A) \cdot \phi(B) = 1$, then $\phi(A) = \phi(B)$, so this could not happen if $\mathcal{K}$ was a kernel.[1]

In the following we will use $\ell(x)$ to denote the label of example x and use $x \sim P$ as shorthand for $(x, \ell(x)) \sim P$.

3. Sufficient Conditions for Learning with Similarity Functions

We now provide a series of sufficient conditions for a similarity function to be useful for learning, leading to our main notion given in Definition 3.

We begin with our first and simplest notion of "good similarity function" that is intuitive and yields an immediate learning algorithm, but which is not broad enough to capture all good kernel functions. Nonetheless, it provides a convenient starting point. This definition says that $\mathcal{K}$ is a good similarity function for a learning problem P if most examples x (at least a $1-\epsilon$ probability mass) are on average at least γ more similar to random examples x' of the *same* label than they are to random examples x' of the opposite label. Formally,

Definition 2 *$\mathcal{K}$ is a* **strongly (ϵ, γ)-good similarity function** *for a learning problem P if at least a $1 - \epsilon$ probability mass of examples x satisfy: $\mathbf{E}_{x' \sim P}[\mathcal{K}(x, x')|\ell(x') = \ell(x)] \geq \mathbf{E}_{x' \sim P}[\mathcal{K}(x, x')|\ell(x') \neq \ell(x)] + \gamma$.*

For example, suppose all positive examples have similarity at least 0.2 with each other, and all negative examples have similarity at least 0.2 with each other, but positive and negative examples have similarities distributed uniformly at random in $[-1, 1]$. Then, this would satisfy Definition 2 for $\gamma = 0.2$ and $\epsilon = 0$, but with high probability would not be positive semidefinite.[2]

Definition 2 captures an intuitive notion of what one might want in a similarity function. In addition, if a similarity function $\mathcal{K}$ satisfies Definition 2 then it suggests a simple, natural learning algorithm: draw a sufficiently large set S^+ of positive examples and set S^- of negative examples, and then output the prediction rule that classifies a new example x as positive if it is on average more similar to points in S^+ than to points in S^-, and negative otherwise. Formally:

Theorem 1 *If $\mathcal{K}$ is strongly (ϵ, γ)-good, then $(4/\gamma^2) \ln(2/\delta)$ positive examples S^+ and $(4/\gamma^2) \ln(2/\delta)$ negative examples S^- are sufficient so that with probability $\geq 1 - \delta$, the above algorithm produces a classifier with error at most $\epsilon + \delta$.*

Proof: Let Good be the set of x satisfying $\mathbf{E}_{x' \sim P}[\mathcal{K}(x, x')|\ell(x) = \ell(x')] \geq \mathbf{E}_{x' \sim P}[\mathcal{K}(x, x')|\ell(x) \neq \ell(x')] + \gamma$. So, by assumption, $\Pr_{x \sim P}[x \in \mathsf{Good}] \geq 1 - \epsilon$. Now, fix $x \in \mathsf{Good}$. Since $\mathcal{K}(x, x') \in [-1, 1]$, by Hoeffding bounds we have that over the random draw of the sample S^+, $\Pr\left(\left|\mathbf{E}_{x' \in S^+}[\mathcal{K}(x, x')] - \mathbf{E}_{x' \sim P}[\mathcal{K}(x, x')|\ell(x') = 1]\right| \geq \gamma/2\right) \leq 2e^{-2|S^+|\gamma^2/4}$, and similarly for S^-. By our choice of $|S^+|$ and $|S^-|$, each of these probabilities is at most $\delta^2/2$.

So, for any given $x \in \mathsf{Good}$, there is at most a δ^2 probability of error over the draw of S^+ and S^-. Since this is true for any $x \in \mathsf{Good}$, it implies that the *expected* error of this procedure, over $x \in \mathsf{Good}$, is at most δ^2, which by Markov's inequality implies that there is at most a δ probability that the error rate over Good is more than δ. Adding in the ϵ probability mass of points not in Good yields the theorem. ■

Theorem 1 implies that if $\mathcal{K}$ is a strongly (ϵ, γ)-good similarity function for small ϵ and not-too-small γ, then it can be used in a natural way for learning. However, Definition 2 is not sufficient to capture all good kernel functions. In particular, Figure 3.1 gives a simple example in R^2 where the standard kernel $\mathcal{K}(x, x') = x \cdot x'$ has a large margin separator (margin of $1/2$) and yet does not satisfy Definition 2, even for $\gamma = 0$ and $\epsilon = 0.49$.

Notice, however, that if in Figure 3.1 we simply ignored the positive examples in the upper-left when choosing x', then we would be fine. In fact, if we were *given* a weighting function w that down-weighted certain regions of the

[1] You could make such a function positive semidefinite by instead defining similarity to be the *number* of authors and keywords in common, but perhaps that is not what you want for the task at hand. Alternatively, you can make the similarity matrix positive semidefinite by blowing up the diagonal, but that would reduce the normalized margin.

[2] In particular, if the domain is large enough, then with high probability there would exist negative example A and positive examples B, C such that $\mathcal{K}(A, B)$ is close to 1 (so they are nearly identical as vectors), $\mathcal{K}(A, C)$ is close to -1 (so they are nearly opposite as vectors), and yet $\mathcal{K}(B, C) \geq 0.2$ (their vectors form an acute angle).

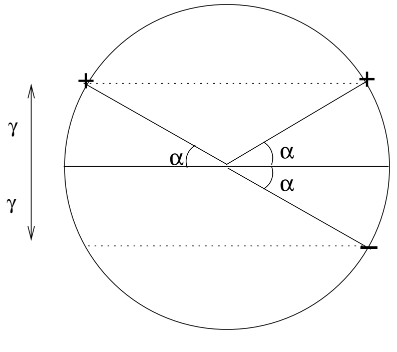

Figure 3.1. Positives are split equally among upper-left and upper-right. Negatives are all in the lower-right. For $\alpha = 30^o$ (so $\gamma = 1/2$) a large fraction of the positive examples (namely the 50% in the upper-right) have a higher dot-product with negative examples ($\frac{1}{2}$) than with a random positive example ($\frac{1}{2} \cdot 1 + \frac{1}{2}(-\frac{1}{2}) = \frac{1}{4}$).

input space, so that at least a $1-\epsilon$ probability mass of examples x satisfy $\mathbf{E}_{x' \sim P}[w(x')\mathcal{K}(x,x')|\ell(x) = \ell(x')] \geq \mathbf{E}_{x' \sim P}[w(x')\mathcal{K}(x,x')|\ell(x) \neq \ell(x')] + \gamma$, then we could use $\mathcal{K}$ and w to learn in exactly the same way as a similarity function $\mathcal{K}$ satisfying Definition 2.[3] This now motivates our main definition given below. The key difference is that whereas in the above observation one would need the designer to construct both the similarity function $\mathcal{K}$ *and* the weighting function w, in Definition 3 we only require that such a w *exist*, but it need not be known a-priori.

Definition 3 (main) *A similarity function $\mathcal{K}$ is an (ϵ, γ)-* **good similarity function** *for a learning problem P if there* exists *a bounded weighting function w over X ($w(x') \in [0,1]$ for all $x' \in X$) such that at least a $1-\epsilon$ probability mass of examples x satisfy: $\mathbf{E}_{x' \sim P}[w(x')\mathcal{K}(x,x')|\ell(x) = \ell(x')] \geq \mathbf{E}_{x' \sim P}[w(x')\mathcal{K}(x,x')|\ell(x) \neq \ell(x')] + \gamma$.*

We now show two interesting properties of Definition 3. First, if $\mathcal{K}$ is a similarity function satisfying it, then we can use $\mathcal{K}$ to explicitly map the data into a space in which there is a separator with low-error (not much more than ϵ) at a large margin (not too much less than γ), and thereby convert the learning problem into a standard one of learning a linear separator. The second is that any "good kernel" (a kernel with a large margin separator in its implicit ϕ-space) must satisfy Definition 3, though with some degradation in the parameters. We prove the first statement, which is the easier of the two, in this section, and we will consider the second one, which has a more involved proof, in Section 4.

Theorem 2 *If $\mathcal{K}$ is an (ϵ, γ)-good similarity function, then if one draws a set S from P containing $d = (4/\gamma)^2 \ln(2/\delta)$ positive examples $S^+ = \{y_1, y_2, \ldots, y_d\}$ and d negative examples $S^- = \{z_1, z_2, \ldots, z_d\}$, then with probability at least $1-\delta$, the mapping $\rho_S : X \to R^{2d}$ defined as $\rho_S(x) = (\mathcal{K}(x,y_1), \ldots, \mathcal{K}(x,y_d), \mathcal{K}(x,z_1), \ldots, \mathcal{K}(x,z_d))$ has the property that the induced distribution $\rho_S(P)$ in R^{2d} has a separator of error at most $\epsilon+\delta$ at margin at least $\gamma/4$.*

Proof: Consider the linear separator $\tilde{w}$ in the ρ_S space defined as $\tilde{w}_i = w(y_i)$, for $i \in \{1, 2, \ldots, d\}$ and $\tilde{w}_i = -w(z_{i-d})$, for $i \in \{d+1, d+2, \ldots, 2d\}$. We will show that, with probability at least $(1-\delta)$, $\tilde{w}$ has error at most $\epsilon + \delta$ at margin $\gamma/4$. Let Good be the set of x satisfying inequality $\mathbf{E}_{x' \sim P}[w(x')\mathcal{K}(x,x')|\ell(x) = \ell(x')] \geq \mathbf{E}_{x' \sim P}[w(x')\mathcal{K}(x,x')|\ell(x) \neq \ell(x')] + \gamma$; so, by assumption, $\Pr_{x \sim P}[x \in \mathsf{Good}] \geq 1 - \epsilon$.

Consider some fixed point $x \in \mathsf{Good}$. We begin by showing that for any such x,

$$\Pr_{S^+,S^-}\left(\ell(x)\frac{\tilde{w} \cdot \rho_S(x)}{||\tilde{w}||\ ||\rho_S(x)||} \geq \frac{\gamma}{4}\right) \geq 1 - \delta^2.$$

To do so, first notice that d is large enough so that with high probability, at least $1-\delta^2$, we have both $|\mathbf{E}_{x' \in S^+}[w(x')\mathcal{K}(x,x')] - \mathbf{E}_{x' \sim P}[w(x')\mathcal{K}(x,x')|\ell(x') = 1]| \leq \frac{\gamma}{4}$ and $|\mathbf{E}_{x' \in S^-}[w(x')\mathcal{K}(x,x')] - \mathbf{E}_{x' \sim P}[w(x')\mathcal{K}(x,x')|\ell(x') = -1]| \leq \frac{\gamma}{4}$. Let's consider now the case when $\ell(x) = 1$. In this case we have $\ell(x)\tilde{w} \cdot \rho_S(x) = d(\frac{1}{d}\sum_{i=1}^d w(y_i)\mathcal{K}(x,y_i) - \frac{1}{d}\sum_{i=1}^d w(z_i)\mathcal{K}(x,z_i))$, and so combining these facts we have that with probability at least $(1-\delta^2)$ the following holds: $\ell(x)\tilde{w} \cdot \rho_S(x) \geq d(\mathbf{E}_{x' \sim P}[w(x')\mathcal{K}(x,x')|\ell(x') = 1] - \gamma/4 - \mathbf{E}_{x' \sim P}[w(x')\mathcal{K}(x,x')|\ell(x') = -1] - \gamma/4)$. Since $x \in \mathsf{Good}$, this then implies that $\ell(x)\tilde{w} \cdot \rho_S(x) \geq d\gamma/2$. Finally, since $w(x') \in [-1,1]$ for all x', and since $\mathcal{K}(x,x') \in [-1,1]$ for all pairs x, x', we have that $||\tilde{w}|| \leq \sqrt{2d}$ and $||\rho_S(x)|| \leq \sqrt{2d}$, which implies $\Pr_{S^+,S^-}\left(\ell(x)\frac{\tilde{w} \cdot \rho_S(x)}{||\tilde{w}||||\rho_S(x)||} \geq \frac{\gamma}{4}\right) \geq 1 - \delta^2$. The same analysis applies for the case that $\ell(x) = -1$.

Since the above holds for any $x \in \mathsf{Good}$, it is also true for random $x \in \mathsf{Good}$, which implies by Markov's inequality that with probability $1-\delta$, the vector $\tilde{w}$ has error at most δ at margin $\gamma/4$ over P restricted to points $x \in \mathsf{Good}$. Adding back the ϵ probability mass of points x not satisfying $\mathbf{E}_{x' \sim P}[w(x')\mathcal{K}(x,x')|\ell(x) = \ell(x')] \geq \mathbf{E}_{x' \sim P}[w(x')\mathcal{K}(x,x')|\ell(x) \neq \ell(x')] + \gamma$, yields the theorem. ■

Theorem 2 states that if $\mathcal{K}$ is a good similarity function then with high probability there exists a low-error (at most $\epsilon + \delta$) large-margin (at least $\frac{\gamma}{4}$) separator in the transformed space under mapping ρ_S. Furthermore the dimensionality of this space is not too large, only $O(\frac{1}{\gamma^2}\log\frac{1}{\delta})$. Thus, all we need now to learn well is to draw a new, fresh sample, map it into the transformed space using ρ_S, and then apply a good algorithm for learning linear separators in the new space.[4]

[3] The proof is similar to that of Theorem 1 (except now we view $w(x')\mathcal{K}(x,x')$ as the bounded random variable we plug into Hoeffding bounds).

[4] One interesting aspect to notice is that we can use *unlabeled examples* instead of labeled examples when defining the mapping

Remark: Standard margin bounds imply that if $\mathcal{K}$ is a good *kernel* function, then with high probability, the points in a random sample $S = \{x_1, \ldots, x_d\}$ can be assigned weights so that the resulting vector defines a low-error large-margin separator in the ϕ-space. Definition 3 can be viewed as requiring that each point x' have a weight $w(x')$ that is solely a function of the example itself and not the set S to which it belongs. The results in Section 4 below imply that if $\mathcal{K}$ is a good kernel, then these "sample independent" weights must exist as well.

3.1. Variations Tailored to Efficient Algorithms

Our implication is that there will exist a low-error large-margin separator in the transformed space, so that we can then run a standard linear-separator learning algorithm. Technically, however, the guarantee for algorithms such as SVM and Perceptron is not that they necessarily find the *minimum-error* separator on their data (which is NP-hard) but rather that they find the separator that minimizes the *total distance* one would need to move points to put them on the correct side by the given margin. Thus, our worst-case guarantee for SVM and Perceptron is only that they find a separator of error $O((\epsilon + \delta)/\gamma)$, though such algorithms are known to do quite well in practice and the same issue would apply for the definition of an (ϵ, γ)-good kernel.[5]

We can also modify our definition to capture the notion of good similarity functions for the SVM and Perceptron algorithms as follows:

Definition 4 (tailored to SVM and Perceptron) *A similarity function $\mathcal{K}$ is an (ϵ, γ)-***good similarity function in hinge loss** *for a learning problem P if there exists a weighting function $w(x') \in [0, 1]$ for all $x' \in X$ such that*

$$\frac{1}{\gamma}\mathbf{E}_x\Big[\max\big(\gamma_x, 0\big)\Big] \leq \epsilon,$$

where $\gamma_x = \mathbf{E}_{x' \sim P}[w(x')\mathcal{K}(x, x')|\ell(x) \neq \ell(x')] + \gamma - \mathbf{E}_{x' \sim P}[w(x')\mathcal{K}(x, x')|\ell(x) = \ell(x')]$.

In other words, we are asking: on average, by how much would a random example x fail to satisfy the desired γ separation between the weighted similarity to examples of its own label and the weighted similarity to examples of the other label (this is γ_x). This quantity is then scaled by $1/\gamma$.

By applying the same analysis as in the proof of Theorem 2, one can show that given a similarity function satisfying this definition, we can use SVM in the transformed space to achieve error $O(\epsilon + \delta)$.

3.2. Combining Multiple Similarity Functions

Suppose that rather than having a single similarity function, we were instead given n functions $\mathcal{K}_1, \ldots, \mathcal{K}_n$, and our hope is that some convex combination of them will satisfy Definition 3. Is this sufficient to be able to learn well? (Note that a convex combination of similarity functions is guaranteed to have range $[-1, 1]$ and so be a legal similarity function.) The following generalization of Theorem 2 shows that this is indeed the case, though the margin parameter drops by a factor of $\sqrt{n}$. This result can be viewed as analogous to the idea of learning a kernel matrix studied by Lanckriet et al. (2004) except that rather than explicitly learning the best convex combination, we are simply folding the learning process into the second stage of the algorithm.

Theorem 3 *Suppose $\mathcal{K}_1, \ldots, \mathcal{K}_n$ are similarity functions such that some (unknown) convex combination of them is (ϵ, γ)-good. If one draws a set S from P containing $d = (4/\gamma)^2 \ln(2/\delta)$ positive examples $S^+ = \{y_1, y_2, \ldots, y_d\}$ and d negative examples $S^- = \{z_1, z_2, \ldots, z_d\}$, then with probability at least $1 - \delta$, the mapping $\rho_S : X \rightarrow R^{2nd}$ defined as $\rho_S(x) = (\mathcal{K}_1(x, y_1), \ldots, \mathcal{K}_n(x, y_d), \mathcal{K}_1(x, z_1), \ldots, \mathcal{K}_n(x, z_d))$ has the property that the induced distribution $\rho_S(P)$ in R^{2nd} has a separator of error at most $\epsilon + \delta$ at margin at least $\gamma/(4\sqrt{n})$.*

Proof: Let $\mathcal{K} = \alpha_1\mathcal{K}_1 + \ldots + \alpha_n\mathcal{K}_n$ be an (ϵ, γ)-good convex-combination of the $\mathcal{K}_i$. By Theorem 2, had we instead performed the mapping: $\hat{\rho}_S : X \rightarrow R^{2d}$ defined as

$$\hat{\rho}_S(x) = (\mathcal{K}(x, y_1), \ldots, \mathcal{K}(x, y_d), \mathcal{K}(x, z_1), \ldots, \mathcal{K}(x, z_d)),$$

then with probability $1 - \delta$, the induced distribution $\hat{\rho}_S(P)$ in R^{2d} would have a separator of error at most $\epsilon + \delta$ at margin at least $\gamma/4$. Let $\hat{w}$ be the vector corresponding to such a separator in that space. Now, let us convert $\hat{w}$ into a vector in R^{2nd} by replacing each coordinate $\hat{w}_j$ with the n values $(\alpha_1\hat{w}_j, \ldots, \alpha_n\hat{w}_j)$. Call the resulting vector $\tilde{w}$. Notice that by design, for any x we have $\tilde{w} \cdot \rho_S(x) = \hat{w} \cdot \hat{\rho}_S(x)$. Furthermore, $||\tilde{w}|| \leq ||\hat{w}||$ (the worst case is when exactly one of the α_i is equal to 1 and the rest are 0). Thus, the vector $\tilde{w}$ under distribution $\rho_S(P)$ has the same properties as the vector $\hat{w}$ under $\hat{\rho}_S(P)$, except that $||\rho_S(x)||$ may now be as large as $\sqrt{2nd}$ rather than the upper-bound of $\sqrt{2d}$ on $||\hat{\rho}_S(x)||$ used in the proof of Theorem 2. Thus, the margin bound drops by a factor of $\sqrt{n}$. ■

Note that the above argument actually shows something a bit stronger than Theorem 3. In particular, if we define $\alpha = (\alpha_1, \ldots, \alpha_n)$ to be the mixture vector for the optimal $\mathcal{K}$, then we can replace the margin bound $\gamma/(4\sqrt{n})$ with $\gamma/(4||\alpha||\sqrt{n})$. For example, if α is the uniform mixture, then we just get the bound in Theorem 2 of $\gamma/4$.

3.3. Multiclass Classification

We can naturally extend all our results to multiclass classification. In particular, the analog of Definition 3 in that

$\rho_S(x)$. However, if the data distribution is highly unbalanced, say with substantially more negatives than positives, then the mapping may no longer have the large-margin property.

[5] Recent results of Kalai et al. (Kalai et al., 2005) give a method to efficiently perform agnostic learning, achieving error rate arbitrarily close to $\epsilon + \delta$, if the distribution of points in the transformed space is sufficiently "well-behaved".

we require most examples to have their average weighted similarity to points of their own class be at least γ greater than there average weighted similarity to *each* of the other classes. One can then learn using standard adaptations of linear-separator algorithms to the multiclass case (e.g., see Freund and Schapire (1999)).

4. Good Kernels are Good Similarity Functions

In this section we show that a good kernel in the standard sense (i.e. a kernel with a large margin separator in its implicit ϕ-space) will also satisfy Definition 3, though with some degradation of the parameters. Formally:

Theorem 4 *If $\mathcal{K}$ is a* (ϵ, γ)-good kernel function, then for any $\epsilon_{acc} > 0$, $\mathcal{K}$ is a $(\frac{8(\epsilon+2\epsilon_{acc})}{\gamma}, \frac{1}{2M(\gamma,\epsilon_{acc})})$-good similarity function, where $M\,(\gamma, \epsilon) = \frac{1}{\epsilon}\left(\frac{3}{\gamma^2} + \log(\frac{1}{\epsilon})\right)$.

For example, if $\mathcal{K}$ is a $(0, \gamma)$-good kernel function, then for any ϵ' it is also a $(\epsilon', \tilde{O}(\epsilon'\gamma^3))$-good similarity function. To prove Theorem 4, we first need some useful lemmas.

4.1. Some Preliminary Lemmas

The first step to proving Theorem 4 is to show the following. Let P be a normalized (ϵ, γ)-linearly separable learning problem. For any $\epsilon_{acc}, \delta > 0$, if we draw a sample $S = \{z_1, ..., z_M\}$ of size at least $M = \frac{1}{\epsilon_{acc}}\left(\frac{3}{\gamma^2} + \log\left(\frac{1}{\delta}\right)\right)$, then with probability at least $(1 - \delta)$ there exists a weighting $\tilde{w}(z_i) \in \{0, 1\}$ of the examples $z_i \in S$ such that the linear separator w_S given by $w_S = \sum_{i=1}^{M} \ell(z_i)\tilde{w}(z_i)z_i$ has length $||w_S|| \leq 3/\gamma$ and error at most $\epsilon + \epsilon_{acc}$ at margin $1/||w_S||$. To prove this, we consider a modification of the standard Perceptron algorithm (Minsky & Papert, 1969; Novikoff, 1962) which we call Margin-Perceptron.[6]

Algorithm 1 Margin-Perceptron

```
Let$\langle(x_1, \ell(x_1)), \ldots, (x_m, \ell(x_m))\rangle$ be the sequence of
labeled examples. Initialize $t := 1$, and start with
$w_1 = \ell(x_1)x_1$.
For $i = 2, \ldots, m$:
    • Predict positive if $w_t \cdot x_i \geq 1$, predict negative if $w_t \cdot x_i \leq -1$,
      and consider an example to be a margin mistake when $w_t \cdot x_i \in (-1, 1)$.
    • On a mistake (incorrect prediction, or margin mistake), update as follows:
      $w_{t+1} := w_t + \ell(x_i)x_i$, $t := t + 1$.
```

We can now prove the following guarantee on the number of updates made by the Margin-Perceptron algorithm.

Lemma 5 *Let $\mathcal{S} = \langle(x_1, \ell(x_1)), \ldots, (x_m, \ell(x_m))\rangle$ be a sequence of labeled examples with $||x_i|| = 1$. Suppose that there exists a vector w^* such that $||w^*|| = 1$ and $\ell(x_i)w^* \cdot x_i \geq \gamma$ for all examples in the sequence $\mathcal{S}$. Then the number of updates N on $\mathcal{S}$ made by the Margin-Perceptron algorithm is at most $3/\gamma^2$, and furthermore $||w_t|| \leq 3/\gamma$ for all t.*

In other words, the mistake-bound is comparable to that of the standard Perceptron algorithm, and in addition the algorithm if cycled through the data produces a hypothesis w_t of margin at least $1/||w_t|| \geq \gamma/3$.

Proof: Let w_k denote the prediction vector used prior to the kth update (here by an update we mean an update on an incorrect prediction or margin mistake). Thus, if the kth update occurs on $(x_i, \ell(x_i))$, then $\ell(x_i)w_k \cdot x_i < 1$ and $w_{k+1} := w_k + \ell(x_i)x_i$.

As in the classical proof of Perceptron algorithm, we analyze $||w_t||$ and $w_t \cdot w^*$. First, we have $w_{t+1} \cdot w^* \geq \gamma t$ since all examples x_i satisfy $\ell(x_i)w^* \cdot x_i \geq \gamma$. Second, since $||w_{k+1}||^2 = ||w_k||^2 + 2\ell(x_i)w_k \cdot x_i + 1 \leq ||w_k||^2 + 3$, after t updates we have $||w_t||^2 \leq 3t$. Putting these together we obtain that the number of updates N satisfies $\gamma N \leq \sqrt{3N}$ and therefore $N \leq 3/\gamma^2$. Furthermore since $||w_t||^2 \leq 3t$ for all t, this means that for all t we have $||w_t|| \leq 3/\gamma$. ■

Using Lemma 5 we can now show the following structural result for (ϵ, γ)-linearly separable learning problems:

Lemma 6 *Let P be a normalized (ϵ, γ)-linearly separable learning problem. Then, for any $\epsilon_{acc}, \delta > 0$, if we draw a sample $S = \{z_1, ..., z_M\}$ of size $M = \frac{1}{\epsilon_{acc}}\left(\frac{3}{\gamma^2} + \log\left(\frac{1}{\delta}\right)\right)$, then with probability at least $(1 - \delta)$ (over the draw of our sample) there exists a weighting of the examples z_i appearing in the sample $\tilde{w}(z_1), ..., \tilde{w}(z_M) \in \{0, 1\}$ with the property that the linear separator w_S given by $w_S = \sum_{i=1}^{M} \ell(z_i)\tilde{w}(z_i)z_i$ has $||w_S|| \leq 3/\gamma$ and error at most $\epsilon + \epsilon_{acc}$ at margin $1/||w_S||$.*

Proof Sketch: We simply use the weighting function given by the Margin Perceptron algorithm when applied only over the "good" points in the sample (those that are correctly separated by margin γ). If our distribution had error 0 at margin γ, then the result would follow from Lemma 5 and a standard result of converting an online mistake bound guarantee into a PAC guarantee (Littlestone, 1989). Adding ϵ probability mass for the points not having margin γ, we get the desired result. ■

4.2. The Main Argument

We can now use Lemma 6 to prove Theorem 4 that a good kernel function is in fact a good similarity function in the sense of Definition 3.

[6] Note: we are using the margin-Perceptron algorithm here only as a proof technique.

Proof of Theorem 4: The proof follows from the definition of a kernel and Theorem 7 below. ■

Theorem 7 *If P is a normalized (ϵ, γ)-linearly separable learning problem, then for any $\epsilon_{acc} > 0$ there exists a weighting $w(x)$ of examples x in X such that $w(x) \in [0,1]$ and at least a $1 - \frac{8(\epsilon + 2\epsilon_{acc})}{\gamma}$ probability mass of the examples x satisfy: $\mathbf{E}_{x' \sim P}[w(x')x' \cdot x | \ell(x) = \ell(x')] \geq \mathbf{E}_{x' \sim P}[w(x')x' \cdot x | \ell(x) \neq \ell(x')] + \frac{1}{2M}$, where $M = \frac{1}{\epsilon_{acc}}\left(\frac{3}{\gamma^2} + \log\left(\frac{1}{\epsilon_{acc}}\right)\right)$.*

Proof Sketch: From Lemma 6 we know that with probability at least $(1 - \delta)$, a randomly drawn sequence S of $M = \frac{1}{\epsilon_{acc}}\left(\frac{3}{\gamma^2} + \log\left(\frac{1}{\delta}\right)\right)$ examples can be given weights in $\{0, 1\}$ so that the resulting classifier w_S has the property that $\Pr_x [\ell(x)(w_S \cdot x) \leq 1] \leq \epsilon + \epsilon_{acc}$ and furthermore $||w_S|| \leq 3/\gamma$. Define $w_{x',S}$ to be the *total* weight given to example x' in S and let w_S be $\sum_{x' \in S} \ell(x') w_{x',S} x'$; so, formally $w_{x',S}$ is a sum of weights $\tilde{w}(x')$ over all appearances of x' in the sample S (if x' appears multiple times), and w_S is a weighted sum of examples in S. We will say that x is bad with respect to w_S if $\ell(x)(w_S \cdot x) \leq 1$.

Now notice that for each sample S of size M the set of "bad" x's with respect to w_S could be the same or different. However, we know that:

$$\Pr_S \left[\Pr_x [\ell(x)(w_S \cdot x) \leq 1] \geq \epsilon + \epsilon_{acc} \right] \leq \delta.$$

This then implies that at most a $\frac{8(\epsilon + \epsilon_{acc} + \delta)}{\gamma}$ probability mass of x's are bad for more than a $\frac{\gamma}{8}$ probability mass of S's. Define Good to be the remainder of the x's. So, for $x \in$ Good we have that over the random draw of S, there is at least a $1 - \gamma/8$ chance that $\ell(x) w_S \cdot x \geq 1$, and in the remaining $\gamma/8$ chance, we at least know that $\ell(x) w_S \cdot x \geq -||w_S|| \geq -3/\gamma$. This implies that for $x \in$ Good, we have $\mathbf{E}_S[\ell(x) w_S \cdot x] \geq (1 - \frac{\gamma}{8})1 - \frac{\gamma}{8}(\frac{3}{\gamma})$ and so for $x \in$ Good we have $\mathbf{E}_S[\ell(x) w_S \cdot x] \geq \frac{1}{2}$.

Consider the following weighting scheme $w(x') = \Pr[\ell(x')]\mathbf{E}_S[w_{x',S} I(x' \in S)]/\mathbf{E}_S[\#(x' \in S)]$, where $\Pr[\ell(x')]$ is the probability mass over examples in P with the same label as x'. Note that $w(x') \in [0, 1]$. We will show that this weighting scheme satisfies the desired property and for simplicity of notation consider a discrete space X. Consider $x \in$ Good. We know that $\mathbf{E}_S[\ell(x) w_S \cdot x] \geq \frac{1}{2}$, and expanding out w_S we get $\ell(x)\mathbf{E}_S[\sum_{x' \in S} \ell(x') w_{x',S} x'] \cdot x \geq \frac{1}{2}$, or $\ell(x)\ell(x') \sum_{x'} (e(x', S)x') \cdot x \geq \frac{1}{2}$, where $e(x', S) = \mathbf{E}_S[w_{x',S} I(x' \in S)]$. This implies that $\ell(x)\ell(x') \sum_{x'} \left(e(x', S) \frac{1}{\mathbf{E}[\#(x' \in S)]} \frac{\mathbf{E}\left[(\# x' \in S)\right]}{M} \right) x' \cdot x \geq \frac{1}{2M}$, and therefore $\ell(x)\mathbf{E}_{x'}\left[\frac{w(x')}{\Pr[\ell(x')]}(x' \cdot x)\right] \geq \frac{1}{2M}$. Thus we have shown that for $x \in$ Good we have: $\mathbf{E}_{x' \sim P}[w(x')x' \cdot x | \ell(x) = \ell(x')] \geq \mathbf{E}_{x' \sim P}[w(x')x' \cdot x | \ell(x) \neq \ell(x')] + \frac{1}{2M}$. Finally, picking $\delta = \epsilon_{acc}$ we obtain the desired result. ■

5. Similarity Functions, Weak Learning, and Kernel-Target Alignment

Our definitions so far have required that almost all of the points (at least a $1 - \epsilon$ fraction) be on average more similar (perhaps in a weighted sense) to random points of the same label than to those of the other label. A weaker notion would be simply to require that two random points of the same label be on average more similar than two random points of different labels. For instance, one could consider the following generalization of Definition 2:

Definition 5 *$\mathcal{K}$ is a* **weakly γ-good similarity function** *for a learning problem P if: $\mathbf{E}_{x,x' \sim P}[\mathcal{K}(x, x') | \ell(x) = \ell(x')] \geq \mathbf{E}_{x,x' \sim P}[\mathcal{K}(x, x') | \ell(x) \neq \ell(x')] + \gamma$.*

While Definition 5 still captures a natural intuitive notion of what one might want in a similarity function, it is not powerful enough to imply *strong* learning unless γ is quite large.[7] We can however show that for any $\gamma > 0$, Definition 5 is enough to imply weak learning (Schapire, 1990). In particular, we can show that the following natural and simple algorithm is sufficient to weak learn: draw a sufficiently large set S^+ of positive examples and set S^- of negative examples. Then, for each x, consider $\tilde{\gamma}(x) = \frac{1}{2}\left[\mathbf{E}_{x' \in S^+}[\mathcal{K}(x, x')] - \mathbf{E}_{x' \in S^-}[\mathcal{K}(x, x')]\right]$, and finally to classify x use the following probabilistic prediction rule: classify x as positive with probability $\frac{1 + \tilde{\gamma}(x)}{2}$ and as negative with probability $\frac{1 - \tilde{\gamma}(x)}{2}$. (Notice that $\tilde{\gamma}(x) \in [-1, 1]$ and so our algorithm is well defined.) Then we can prove that:

Theorem 8 *If $\mathcal{K}$ is a weakly γ-good similarity function, then if one draws a set S from P containing at least $\frac{32}{\gamma^2} \ln\left(\frac{8}{\gamma\delta}\right)$ positive examples S^+ and at least $\frac{32}{\gamma^2} \ln\left(\frac{8}{\gamma\delta}\right)$ negative examples S^-, then with probability at least $1 - \delta$, the above probabilistic classifier has error at most $\frac{1}{2} - \frac{7\gamma}{128}$.*

Proof: Omitted. ■

5.1. Relationship to Kernel Target Alignment

It is interesting to notice the close relationship between Definition 5 and the notion of *Kernel Target Alignment* of Cristianini et al. (2001). Specifically, the alignment between a normalized kernel function $\mathcal{K}$ and the target $\ell(x)$ is defined as $A(\mathcal{K}, l(x)) = \mathbf{E}_{x,x' \sim P}[\ell(x)\ell(x')\mathcal{K}(x, x')]$. Notice that this is essentially the same as Definition 5 when the distribution P is balanced among positive and negative examples. As pointed out in (Cristianini et al.,

[7] For example, suppose the instance space is the real line and that the similarity measure $\mathcal{K}$ we are considering is the standard dot product: $\mathcal{K}(x, x') = x \cdot x'$. Assume the distribution is 50% positive, 50% negative, and that 75% of the negative examples are at position -1 and 25% are at position 1, and vice-versa for the positive examples. Then $\mathcal{K}$ is a weakly γ-good similarity function for $\gamma = 1/2$, but the best accuracy one can hope for in this situation is 75%.

2001), if this alignment is very large, then the function $f(x) = \mathbf{E}_{x' \sim P}[l(x')\mathcal{K}(x,x')]$ has high generalization accuracy. In fact, we can get an efficient algorithm in this case using the same approach as in the proof of Theorem 1. One nice feature about our Definitions 2 and 3, however, is that they allow for similarity functions with fairly small values of γ to still produce high-accuracy learning algorithms. For instance, the example given in Section 3 of a similarity function such that all positive examples have similarity at least 0.2 with each other, all negative examples have similarity at least 0.2 with each other, but positives and negatives have similarities uniformly distributed in $[-1,1]$, satisfies Definition 2 with $\epsilon = 0$ and $\gamma = 0.2$. So, using Theorem 1 we can achieve arbitrarily low error. However, for a balanced distribution of positives and negatives (each with 50% probability mass), such a similarity function would have alignment score only 0.2. So, the accuracy achievable based on only using the alignment score would be much lower.

6. Conclusions

The main contribution of this work is to develop a theory of learning with similarity functions: namely, of when a similarity function is good for a given learning problem, that is more general and in terms of more tangible quantities than the standard theory of kernel functions. We provide a definition that we show is both sufficient for learning and satisfied by the usual large-margin notion of a good kernel. Moreover, the similarity properties we consider do not require reference to implicit high-dimensional spaces nor do they require that the similarity function be positive semidefinite. In this way, we provide the first rigorous explanation showing why a kernel function that is good in the large-margin sense can also formally be viewed as a good similarity function, thereby giving formal justification to the standard intuition about kernels. Our results also suggest a possible direction for improved definitions in the context of Kernel-Target Alignment.

6.1. Open Problems and Future Work

While we can show that a kernel $\mathcal{K}$ that has the large margin property in its implicit space is also a good similarity function under our definitions, our reduction results in a loss in the parameters. For example if $\mathcal{K}$ is a $(0,\gamma)$-good kernel function, then using our reduction it is roughly an $(\epsilon, \epsilon\gamma^3)$-good similarity function. One open problem is whether one can improve the argument and the resulting bounds.

Our algorithms (much like those of Balcan et al. (2004)) also suggest a natural way to use kernels or other similarity functions in learning problems for which one also wishes to use the native features of the examples. For instance, consider the problem of classifying a stream of documents arriving one at a time. Rather than running a kernelized learning algorithm, one can simply take the native features (say the words in the document) and augment them with a small number of additional features representing the similarity of the current example with each of a pre-selected set of initial documents. One can then feed the augmented example into a standard unkernelized online learning algorithm. It would be interesting to explore this idea further.

Finally, our results suggest an approach to analyzing similarity functions in the context of clustering. That is, one would ask what properties of pairwise similarity functions are sufficient to allow an algorithm to *cluster* well.

Acknowledgements: This work was supported in part by the National Science Foundation under grants CCF-0514922, CCR-0122581, and IIS-0121678.

References

Anthony, M., & Bartlett, P. (1999). *Neural Network Learning: Theoretical Foundations*. Cambridge University Press.

Balcan, M.-F., Blum, A., & Vempala, S. (2004). On kernels, margins and low-dimensional mappings. *International Conference on Algorithmic Learning Theory*.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*, 273 – 297.

Cristianini, N., Shawe-Taylor, J., Elisseeff, A., & Kandola, J. (2001). On kernel target alignment. *Advances in Neural Information Processing Systems*.

Freund, Y., & Schapire, R. E. (1999). Large margin classification using the perceptron algorithm. *Machine Learning*, *37*, 277 – 296.

Herbrich, R. (2002). *Learning kernel classifiers*. MIT Press.

Joachims, T. (2002). *Learning to classify text using support vector machines: Methods, theory, and algorithms*. Kluwer.

Kalai, A., Klivans, A., Mansour, Y., & Servedio, R. (2005). Agnostically learning halfspaces. *Proceedings of the 46th Annual Symposium on the Foundations of Computer Science*.

Lanckriet, G. R. G., Cristianini, N., Bartlett, P. L., Ghaoui, L. E., & Jordan, M. I. (2004). Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 27–72.

Littlestone, N. (1989). From online to batch learning. *Proc. 2nd ACM Conf. on Computational Learning Theory* (pp. 269–284).

Minsky, M., & Papert, S. (1969). *Perceptrons: An introduction to computational geometry*. The MIT Press.

Muller, K. R., Mika, S., Ratsch, G., Tsuda, K., & Scholkopf, B. (2001). An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, *12*, 181 – 201.

Novikoff, A. B. J. (1962). On convergence proofs on perceptrons. *Proc. Symposium on the Mathematical Theory of Automata*.

Scholkopf, B., Tsuda, K., & Vert, J.-P. (2004). *Kernel methods in computational biology*. MIT Press.

Schapire, R. (1990). The strength of weak learnability. *Machine Learning*, *5*, 197–227.

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

Shawe-Taylor, J., & Cristianini, N. (2004). *Kernel methods for pattern analysis*. Cambridge University Press.

Smola, A. J., Bartlett, P., Scholkopf, B., & Schuurmans, D. (2000). *Advances in large margin classifiers*. MIT Press.

Vapnik, V. N. (1998). *Statistical learning theory*. Wiley & Sons.