A Theory of Learning with Similarity Functions[⋆]

Maria-Florina Balcan[1], Avrim Blum[1], and Nathan Srebro[2]

[1] Computer Science Department, Carnegie Mellon University
{ninamf,avrim}@cs.cmu.edu
[2] Toyota Technological Institute at Chicago
nati@uchicago.edu

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 necessarily correspond to the intuition of a good kernel as a good measure of similarity, and the underlying margin in the implicit space usually is not apparent in "natural" representations of the data. Therefore, it may be difficult for a domain expert to use the theory to help design an appropriate kernel for the learning task at hand. Moreover, 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). Instead, our theory talks in terms of more direct properties of how the function behaves as a similarity measure. 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 and more general similarity functions that describes the effectiveness of a given 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 [1, 22, 20, 9, 12, 26]. 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 K, there must exist an (implicit) mapping ϕ, such that for all inputs x, x' we have $K(x, x') = \langle \phi(x), \phi(x') \rangle$. The kernel is then used inside a "kernelized" learning algorithm such as SVM or kernel-perceptron in place of direct access to the data.

The theory behind kernel functions is based on the fact that many standard algorithms for learning linear separators, such as SVMs [26] and the Perceptron [8] 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 $\langle \phi(x), \phi(x') \rangle$ with a kernel computation $K(x, x')$, 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 learning guarantees that depend only on the *margin* of the best separator, and not on the dimension of the space in which the data resides [2, 21]. Thus, kernel functions are often viewed as providing much of the power of this implicit high-dimensional space, without paying for it either 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. When designing a kernel function for some learning problem, the intuition employed typically does not involve implicit high-dimensional spaces but rather that a good kernel would be one that serves as a good measure of similarity for the given problem [20]. In-fact, many generic kernels (e.g. Gaussian kernels), as well as very specific kernels (e.g. Fisher kernels [11] and kernels for specific

[⋆] This paper combines results appearing in two conference papers: M.F. Balcan and A. Blum, "On a Theory of Learning with Similarity Functions," Proc. 23rd International Conference on Machine Learning, 2006 [4], and N. Srebro, "How Good is a Kernel as a Similarity Function," Proc. 20th Annual Conference on Learning Theory, 2007 [24].

structures such as [27]), describe different notions of similarity between objects, which do not correspond to any intuitive or easily interpretable high-dimensional representation. So, in this sense the theory is not always helpful in providing intuition when selecting or designing a kernel function for a particular learning problem. Additionally, 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 the function, to coerce it into a "legal" form. Finally, 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 $K(x, x')$ 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) generalizes the standard theory in that a good kernel in the usual 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.

Our Results: Our main contribution is the development of a theory for what it means for a pairwise function to be a "good similarity function" for a given learning problem, along with theorems showing that our main definitions are sufficient to be able to learn well and in addition generalize the standard notion of a good kernel function, though with some bounded degradation of learning guarantees. We begin with a definition (Definition 4) that is especially intuitive and allows for learning via a very simple algorithm, but is not broad enough to include all kernel functions that induce large-margin separators. We then broaden this notion to our main definition (Definition 8) that requires a more involved algorithm to learn, but is now able to capture all functions satisfying the usual notion of a good kernel function. Specifically, we show that if K is a similarity function satisfying Definition 8 then one can algorithmically perform a simple, *explicit* transformation of the data under which there is a low-error large-margin separator. We also consider variations on this definition (e.g., Definition 9) that produce better guarantees on the quality of the final hypothesis when combined with existing learning algorithms.

A similarity function K satisfying our definition, but that is not positive semi-definite, is not necessarily guaranteed to work well when used directly in standard learning algorithms such as SVM or the Perceptron algorithm[3]. Instead, what we show is that such a similarity function can be employed in the following two-stage algorithm. First, re-represent that data by performing what might be called an "empirical similarity map": selecting a subset of data points as landmarks, and then representing each data point using the similarities to those landmarks. Then, use standard methods to find a large-margin linear separator in the new space. One property of this approach 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 "kernalizable". In fact, this work is motivated by work on a re-representation method that algorithmically transforms a kernel-based learning problem (with a valid positive-semidefinite kernel) to an explicit low-dimensional learning problem [5].

More generally, our framework provides a formal way to analyze properties of a similarity function that make it sufficient for learning, as well as what algorithms are suited for a given property. While our work is motivated by extending the standard large-margin notion of a good kernel function, we expect one can use this framework to analyze other, not necessarily comparable, properties that are sufficient for learning as well. In fact, recent work along these lines is given in [28].

2 Background and Notation

We consider a learning problem specified as follows. We are given access to labeled examples (x, y) 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

[3] However, as we will see in Section 4.2, if the function *is* positive semi-definite and if it is good in our sense, then we can show it is good as a kernel as well.

produce a classification function $g : X \to \{-1, 1\}$ whose error rate $\Pr_{(x,y)\sim P}[g(x) \neq y]$ is low. We will consider learning algorithms that only access the points x through a pairwise similarity function $K(x, x')$ mapping pairs of points to numbers in the range $[-1, 1]$. Specifically,

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

Our goal is to describe "goodness" properties that are sufficient for a similarity function to allow one to learn well that ideally are intuitive and subsume the usual notion of good kernel function. Note that as with the theory of kernel functions [19], "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 [25, 14].

A similarity function K is a valid kernel function if it is positive-semidefinite, i.e. there exists a function ϕ from the instance space X into some (implicit) Hilbert "ϕ-space" such that $K(x, x') = \langle\phi(x), \phi(x')\rangle$. See, e.g., Smola and Schölkopf [23] for a discussion on conditions for a mapping being a kernel function. Throughout this work, and without loss of generality, we will only consider kernels such that $K(x, x) \leq 1$ for all $x \in \mathcal{X}$ (any kernel K can be converted into this form by, for instance, defining $\tilde{K}(x, x') = K(x, x')/\sqrt{K(x, x)K(x', x')}$). We say that K is (ϵ, γ)*-kernel good* for a given learning problem P if there exists a vector β in the ϕ-space that has error ϵ at margin γ; for simplicity we consider only separators through the origin. Specifically:

Definition 2. *K is (ϵ, γ)*-kernel good *if there exists a vector β, $\|\beta\| \leq 1$ such that*

$$\Pr_{(x,y)\sim P}[y\langle\phi(x), \beta\rangle \geq \gamma] \geq 1 - \epsilon.$$

We say that K is γ*-kernel good* if it is (ϵ, γ)*-kernel good* for $\epsilon = 0$; i.e., it has zero error at margin γ.

Given a kernel that is (ϵ, γ)-kernel-good for some learning problem P, a predictor with error rate at most $\epsilon + \epsilon_{\text{acc}}$ can be learned (with high probability) from a sample of[4] $\tilde{\mathcal{O}}\big((\epsilon + \epsilon_{\text{acc}})/(\gamma^2\epsilon_{\text{acc}}^2)\big)$ examples (drawn independently from the source distribution) by minimizing the number of margin γ violations on the sample [17]. However, minimizing the number of margin violations on the sample is a difficult optimization problem [2, 3]. Instead, it is common to minimize the so-called *hinge loss* relative to a margin.

Definition 3. *We say that K is (ϵ, γ)-kernel good* in hinge-loss *if there exists a vector β, $\|\beta\| \leq 1$ such that*

$$\mathbf{E}_{(x,y)\sim P}[[1 - y\langle\beta, \phi(x)\rangle/\gamma]_+] \leq \epsilon,$$

where $[1 - z]_+ = \max(1 - z, 0)$ is the hinge loss.

Given a kernel that is (ϵ, γ)-kernel-good in hinge-loss, a predictor with error rate at most $\epsilon + \epsilon_{\text{acc}}$ can be efficiently learned (with high probability) from a sample of $\mathcal{O}\big(1/(\gamma^2\epsilon_{\text{acc}}^2)\big)$ examples by minimizing the average hinge loss relative to margin γ on the sample [7].

Clearly, a general similarity function might not be a legal kernel. For example, suppose we consider two documents to have similarity 1 if they have either an author in common or a keyword in common, and similarity 0 otherwise. Then you could have three documents A, B, and C, such that $K(A, B) = 1$ because A and B have an author in common, $K(B, C) = 1$ because B and C have a keyword in common, but $K(A, C) = 0$ because A and C have neither an author nor a keyword in common (and $K(A, A) = K(B, B) = K(C, C) = 1$). On the other hand, a kernel requires that if $\phi(A)$ and $\phi(B)$ are of unit length and $\langle\phi(A), \phi(B)\rangle = 1$, then $\phi(A) = \phi(B)$, so this could not happen if K was a valid kernel. Of course, one could modify such a function to be positive semidefinite by, e.g., instead defining similarity to be the *number* of authors and keywords in common, but perhaps that is not the most natural similarity measure for the task at hand. Alternatively, one could make the similarity function positive semidefinite by blowing up the diagonal, but this can significantly decrease the "dynamic range" of K and yield a very small margin.

Deterministic Labels: For simplicity in presentation of our framework, for most of this paper we will consider only learning problems where the label y is a deterministic function of x. For such learning problems, we can use $y(x)$ to denote the label of point x, and we will use $x \sim P$ as shorthand for $(x, y(x)) \sim P$. We will return to learning problems where the label y may be a probabilistic function of x in Section 5.

[4] The $\tilde{\mathcal{O}}(\cdot)$ notations hide logarithmic factors in the arguments, and in the failure probability.

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 notions given in Definitions 8 and 9.

3.1 Simple Sufficient Conditions

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 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 4. *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:*

$$\mathbf{E}_{x'\sim P}[K(x,x')|y(x)=y(x')] \geq \mathbf{E}_{x'\sim P}[K(x,x')|y(x)\neq y(x')]+\gamma. \tag{3.1}$$

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 4 for $\gamma = 0.2$ and $\epsilon = 0$. Note that with high probability this would not be positive semidefinite.[5]

Definition 4 captures an intuitive notion of what one might want in a similarity function. In addition, if a similarity function K satisfies Definition 4 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 K is strongly (ϵ,γ)-good, then $(16/\gamma^2)\ln(2/\delta)$ positive examples S^+ and $(16/\gamma^2)\ln(2/\delta)$ 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 $\mathbf{E}_{x'\sim P}[K(x,x')|y(x)=y(x')] \geq \mathbf{E}_{x'\sim P}[K(x,x')|y(x)\neq y(x')]+\gamma$. So, by assumption, $\Pr_{x\sim P}[x\in \mathsf{Good}] \geq 1-\epsilon$. Now, fix $x \in \mathsf{Good}$. Since $K(x,x')\in[-1,1]$, by Hoeffding bounds we have that over the random draw of the sample S^+, $\Pr\left(\left|\mathbf{E}_{x'\in S^+}[K(x,x')] - \mathbf{E}_{x'\sim P}[K(x,x')|y(x')=1]\right| \geq \gamma/2\right) \leq 2e^{-2|S^+|\gamma^2/16}$, 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. □

Definition 4 requires that almost all of the points (at least a $1-\epsilon$ fraction) be on average more similar to random points of the same label than to random points 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 4:

Definition 5. *K is a* **weakly** γ**-good similarity function** *for a learning problem P if:*

$$\mathbf{E}_{x,x'\sim P}[K(x,x')|y(x)=y(x')] \geq \mathbf{E}_{x,x'\sim P}[K(x,x')|y(x)\neq y(x')]+\gamma. \tag{3.2}$$

[5] 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 $K(A,B)$ is close to 1 (so they are nearly identical as vectors), $K(A,C)$ is close to -1 (so they are nearly opposite as vectors), and yet $K(B,C)\geq 0.2$ (their vectors form an acute angle).

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. For example, suppose the instance space is R^2 and that the similarity measure K we are considering is just the product of the first coordinates (i.e., dot-product but ignoring the second coordinate). Assume the distribution is half positive and half negative, and that 75% of the positive examples are at position $(1, 1)$ and 25% are at position $(-1, 1)$, and 75% of the negative examples are at position $(-1, -1)$ and 25% are at position $(1, -1)$. Then K is a weakly γ-good similarity function for $\gamma = 1/2$, but the best accuracy one can hope for using K is 75% because that is the accuracy of the Bayes-optimal predictor given only the first coordinate.

We can however show that for any $\gamma > 0$, Definition 5 is enough to imply weak learning [18]. In particular, the following simple algorithm is sufficient to weak learn. First, determine if the distribution is noticeably skewed towards positive or negative examples: if so, weak-learning is immediate (output all-positive or all-negative respectively). Otherwise, draw a sufficiently large set S^+ of positive examples and set S^- of negative examples. Then, for each x, consider $\tilde{\gamma}(x) = \frac{1}{2}\left[\mathbf{E}_{x' \in S^+}[K(x, x')] - \mathbf{E}_{x' \in S^-}[K(x, x')]\right]$. 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.) We can then prove the following result:

Theorem 2. *If K is a weakly γ-good similarity function, then with probability at least $1 - \delta$, the above algorithm using sets S^+, S^- of size $\frac{64}{\gamma^2}\ln\left(\frac{64}{\gamma\delta}\right)$ yields a classifier with error at most $\frac{1}{2} - \frac{3\gamma}{128}$.*

Proof. See Appendix A. □

Returning to Definition 4, Theorem 1 implies that if 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 4 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 $K(x, x') = \langle x, x' \rangle$ has a large margin separator (margin of $1/2$) and yet does not satisfy Definition 4, even for $\gamma = 0$ and $\epsilon = 0.49$.

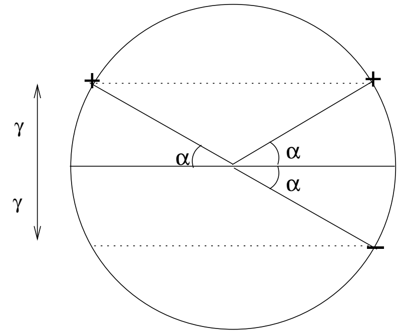

Fig. 3.1. Positives are split equally among upper-left and upper-right. Negatives are all in the lower-right. For $\alpha = 30^o$ (so $\gamma = 1/2$) a large fraction of the positive examples (namely the 50% in the upper-right) have a higher dot-product with negative examples $(\frac{1}{2})$ than with a random positive example $(\frac{1}{2} \cdot 1 + \frac{1}{2}(-\frac{1}{2}) = \frac{1}{4})$. However, if we assign the positives in the upper-left a weight of 0, those in the upper-right a weight of 1, and assign negatives a weight of $\frac{1}{2}$, then all examples have higher average *weighted* similarity to those of the same label than to those of the opposite label, by a gap of $\frac{1}{4}$.

Notice, however, that if in Figure 3.1 we simply ignored the positive examples in the upper-left when choosing x', and down-weighted the negative examples a bit, then we would be fine. This then motivates the following intermediate notion of a similarity function K being good under a weighting function w over the input space that can downweight certain portions of that space.

Definition 6. *A similarity function K together with a bounded weighting function w over X (specifically, $w(x') \in [0, 1]$ for all $x' \in X$) is a* **strongly** (ϵ, γ)**-good weighted similarity function** *for a learning problem P if at least a*

$1-\epsilon$ *probability mass of examples x satisfy:*

$$\mathbf{E}_{x'\sim P}[w(x')K(x,x')|y(x)=y(x')] \geq \mathbf{E}_{x'\sim P}[w(x')K(x,x')|y(x)\neq y(x')]+\gamma. \tag{3.3}$$

We can view Definition 6 intuitively as saying that we only require most examples be substantially more similar on average to *reasonable* points of the same class than to *reasonable* points of the opposite class, where "reasonableness" is a score in $[0,1]$ given by the weighting function w. A pair (K,w) satisfying Definition 6 can be used in exactly the same way as a similarity function K satisfying Definition 4, with the exact same proof used in Theorem 1 (except now we view $w(y)K(x,x')$ as the bounded random variable we plug into Hoeffding bounds).

Unfortunately, Definition 6 requires the designer to construct both K and w, rather than just K. We now weaken the requirement to ask only that such a w *exist*, in Definition 7 below:

Definition 7 (Provisional). *A similarity function 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:*

$$\mathbf{E}_{x'\sim P}[w(x')K(x,x')|y(x)=y(x')] \geq \mathbf{E}_{x'\sim P}[w(x')K(x,x')|y(x)\neq y(x')]+\gamma. \tag{3.4}$$

As mentioned above, the key difference is that whereas in Definition 6 one needs the designer to construct both the similarity function K *and* the weighting function w, in Definition 7 we only require that such a w *exist*, but it need not be known a-priori. That is, we ask only that there exist a large probability mass of "reasonable" points (a weighting scheme) satisfying Definition 6, but the designer need not know in advance what that weighting scheme should be.

Definition 7, which was the main Definition analyzed by Balcan and Blum [4], can also be stated as requiring that, for at least $1-\epsilon$ of the examples, the *classification margin*

$$\begin{aligned}\mathbf{E}_{x'\sim P}[w(x')y(x')K(x,x')|y(x)=y(x')]-\mathbf{E}_{x'\sim P}[w(x')y(x')K(x,x')|y(x)\neq y(x')]\\ =y(x)\mathbf{E}_{x'\sim P}[w(x')y(x')K(x,x')/P(y(x'))]\end{aligned} \tag{3.5}$$

be at least γ, where $P(y(x'))$ is the marginal probability under P, i.e. the prior, of the label associated with x'. We will find it more convenient in the following to analyze instead a slight variant, dropping the factor $1/P(y(x'))$ from the classification margin (3.5)—see Definition 8 in the next Section. For a balanced distribution of positives and negatives (each with 50% probability mass), these two notions are identical, except for a factor of two.

3.2 Main Conditions

We are now ready to present our main sufficient condition for learning with similarity functions. This is essentially a restatement of Definition 7, dropping the normalization by the label "priors" as discussed at the end of the preceding Section.

Definition 8 (Main, Margin Violations). *A similarity function 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:*

$$\mathbf{E}_{x'\sim P}[y(x)y(x')w(x')K(x,x')]\geq\gamma. \tag{3.6}$$

We would like to establish that the above condition is indeed sufficient for learning. I.e. that given an (ϵ,γ)-good similarity function K for some learning problem P, and a sufficiently large labeled sample drawn from P, one can obtain (with high probability) a predictor with error rate arbitrarily close to ϵ. To do so, we will show how to use an (ϵ,γ)-good similarity function K, and a sample S drawn from P, in order to construct (with high probability) an explicit mapping $\phi^S : X\rightarrow\mathbb{R}^d$ for all points in X (not only points in the sample S), such that the mapped data $(\phi^S(x),y(x))$, where $x\sim P$, is separated with error close to ϵ (and in fact also large margin) in the low-dimensional linear space $\mathbb{R}^d$ (Theorem 3 below). We thereby convert the learning problem into a standard problem of learning a linear separator, and can use standard results on learnability of linear separators to establish learnability of our original learning problem, and even provide learning guarantees.

What we are doing is actually showing how to use a good similarity function K (that is not necessarily a valid kernel) and a sample S drawn from P to construct a valid kernel $\tilde{K}^S$, given by $\tilde{K}^S(x, x') = \langle \phi^S(x), \phi^S(x') \rangle$, that is kernel-good and can thus be used for learning (In Section 4 we show that if K is already a valid kernel, a transformation is not necessary as K itself is kernel-good). We are therefore leveraging here the established theory of linear, or kernel, learning in order to obtain learning guarantees for similarity measures that are not valid kernels.

Interestingly, in Section 4 we also show that any kernel that is kernel-good is also a good similarity function (though with some degradation of parameters). The suggested notion of "goodness" (Definition 8) thus encompasses the standard notion of kernel-goodness, and extends it also to non-positive-definite similarity functions.

Theorem 3. *Let K be an (ϵ, γ)-good similarity function for a learning problem P. For any $\delta > 0$, let $S = \{\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_d\}$ be a sample of size $d = 8\log(1/\delta)/\gamma^2$ drawn from P. Consider the mapping $\phi^S : X \to \mathbb{R}^d$ defined as follows: ${\phi^S}_i(x) = \frac{K(x,\tilde{x}_i)}{\sqrt{d}}$, $i \in \{1, \ldots, d\}$. With probability at least $1 - \delta$ over the random sample S, the induced distribution $\phi^S(P)$ in R^d has a separator of error at most $\epsilon + \delta$ at margin at least $\gamma/2$.*

Proof. Let $w : X \to [0, 1]$ be the weighting function achieving (3.6) of Definition 8. Consider the linear separator $\beta \in \mathbb{R}^d$, given by $\beta_i = \frac{y(\tilde{x}_i)w(\tilde{x}_i)}{\sqrt{d}}$; note that $\|\beta\| \le 1$. We have, for any $x, y(x)$:

$$y(x)\langle \beta, \phi^S(x) \rangle = \frac{1}{d}\sum_{i=1}^{d} y(x)y(\tilde{x}_i)w(\tilde{x}_i)K(x, \tilde{x}_i) \tag{3.7}$$

The right hand side of the (3.7) is an empirical average of $-1 \le y(x)y(x')w(x')K(x, x') \le 1$, and so by Hoeffding's inequality, for any x, and with probability at least $1 - \delta^2$ over the choice of S, we have:

$$\frac{1}{d}\sum_{i=1}^{d} y(x)y(\tilde{x}_i)w(\tilde{x}_i)K(x, \tilde{x}_i) \ge \mathbf{E}_{x'\sim P}[y(x)y(x')w(x')K(x, x')] - \sqrt{\frac{2\log(\frac{1}{\delta^2})}{d}} \tag{3.8}$$

Since the above holds for any x with probability at least $1 - \delta^2$ over the choice of S, it also holds with probability at least $1 - \delta^2$ over the choice of x and S. We can write this as:

$$\mathbf{E}_{S\sim P^d}\left[\Pr_{x\sim P}(\text{ violation })\right] \le \delta^2 \tag{3.9}$$

where "violation" refers to violating (3.8). Applying Markov's inequality we get that with probability at least $1 - \delta$ over the choice of S, at most δ fraction of points violate (3.8). Recalling Definition 8, at most an additional ϵ fraction of the points violate (3.6). But for the remaining $1 - \epsilon - \delta$ fraction of the points, for which both (3.8) and (3.6) hold, we have: $y(x)\langle \beta, \phi^S(x) \rangle \ge \gamma - \sqrt{\frac{2\log(\frac{1}{\delta^2})}{d}} = \gamma/2$, where to get the last inequality we use $d = 8\log(1/\delta)/\gamma^2$. □

In order to learn a predictor with error rate at most $\epsilon + \epsilon_{\text{acc}}$ we can set $\delta = \epsilon_{\text{acc}}/2$, draw a sample of size $d = (4/\gamma)^2 \ln(4/\epsilon_{\text{acc}})$ and construct ϕ^S as in Theorem 3. We can now draw a new, fresh, sample, map it into the transformed space using ϕ^S, and then learn a linear separator in the new space. The number of landmarks is dominated by the $\tilde{\mathcal{O}}\big((\epsilon + \epsilon_{\text{acc}})d/\epsilon_{\text{acc}}^2\big) = \tilde{\mathcal{O}}\big((\epsilon + \epsilon_{\text{acc}})/(\gamma^2\epsilon_{\text{acc}}^2)\big)$ sample complexity of the linear learning, yielding the same order sample complexity as in the kernel-case for achieving error at most $\epsilon + \epsilon_{\text{acc}}$: $\tilde{\mathcal{O}}\big((\epsilon + \epsilon_{\text{acc}})/(\gamma^2\epsilon_{\text{acc}}^2)\big)$.

Unfortunately, the above sample complexity refers to learning by finding a linear separator minimizing the error over the training sample. This minimization problem is NP-hard [2], and even NP-hard to approximate [3]. In certain special cases, such as if the induced distribution $\phi^S(P)$ happens to be log-concave, efficient learning algorithms exist [13]. However, as discussed earlier, in the more typical case, one minimizes the *hinge-loss* instead of the number of errors. We therefore consider also a modification of our definition that captures the notion of good similarity functions for the SVM and Perceptron algorithms as follows:

Definition 9 (Main, Hinge Loss). *A similarity function 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*

$$\mathbf{E}_x\Big[\,[1 - y(x)g(x)/\gamma]_+\,\Big] \le \epsilon, \tag{3.10}$$

where $g(x) = \mathbf{E}_{x' \sim P}[y(x')w(x')K(x,x')]$ *is the similarity-based prediction made using* $w()$*, and recall that* $[1 - z]_+ = \max(0, 1-z)$ *is the hinge-loss.*

In other words, we are asking: on average, by how much, in units of γ, 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.

Similarly to Theorem 3, we have:

Theorem 4. *Let* K *be an* (ϵ, γ)*-good similarity function in hinge loss for a learning problem* P*. For any* $\epsilon_1 > 0$ *and* $0 < \delta < \gamma\epsilon_1/4$ *let* $S = \{\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_d\}$ *be a sample of size* $d = 16\log(1/\delta)/(\epsilon_1\gamma)^2$ *drawn from* P*. With probability at least* $1-\delta$ *over the random sample* S*, the induced distribution* $\phi^S(P)$ *in* R^d*, for* ϕ^S *as defined in Theorem 3, has a separator achieving hinge-loss at most* $\epsilon + \epsilon_1$ *at margin* γ*.*

Proof. Let $w : X \to [0,1]$ be the weighting function achieving an expected hinge loss of at most ϵ at margin γ, and denote $g(x) = \mathbf{E}_{x' \sim P}[y(x')w(x')K(x,x')]$. Defining β as in Theorem 3 and following the same arguments we have that with probability at least $1-\delta$ over the choice of S, at most δ fraction of the points x violate 3.8. We will only consider such samples S. For those points that do not violate (3.8) we have:

$$[1 - y(x)\langle \beta, \phi^S(x)\rangle/\gamma]_+ \le [1 - y(x)g(x)/\gamma]_{+}\frac{1}{\gamma}\sqrt{\frac{2\log(\frac{1}{\delta^2})}{d}} \le [1 - y(x)g(x)/\gamma]_+ + \epsilon_1/2 \tag{3.11}$$

For points that do violate (3.8), we will just bound the hinge loss by the maximum possible hinge-loss:

$$[1 - y(x)\langle \beta, \phi^S(x)\rangle/\gamma]_+ \le 1 + \max_x \left|y(x)\left\|\beta\right\|\left\|\phi^S(x)\right\|\right|/\gamma \le 1 + 1/\gamma \le 2/\gamma \tag{3.12}$$

Combining these two cases we can bound the expected hinge-loss at margin γ:

$$\begin{aligned}\mathbf{E}_{x\sim P}\left[[1 - y(x)\langle \beta, \phi^S(x)\rangle/\gamma]_+\right] &\le \mathbf{E}_{x\sim P}[[1 - y(x)g(x)/\gamma]_+] + \epsilon_1/2 + \Pr(\text{violation}\,)\cdot(2/\gamma)\\ &\le \mathbf{E}_{x\sim P}[[1 - y(x)g(x)/\gamma]_+] + \epsilon_1/2 + 2\delta/\gamma\\ &\le \mathbf{E}_{x\sim P}[[1 - y(x)g(x)/\gamma]_+] + \epsilon_1, \qquad (3.13)\end{aligned}$$

where the last inequality follows from $\delta < \epsilon_1\gamma/4$. □

Following the same approach as that suggested following Theorem 3, and noticing that the dimensionality d of the linear space created by ϕ^S is polynomial in $1/\gamma$, $1/\epsilon_1$ and $\log(1/\delta)$, if a similarity function K is a (ϵ,γ)-good similarity function in hinge loss, one can apply Theorem 4 and then use an SVM solver in the ϕ^S-space to obtain (with probability at least $1-\delta$) a predictor with error rate $\epsilon + \epsilon_1$ using $\tilde{O}\big(1/(\gamma^2\epsilon_{\text{acc}}^2)\big)$ examples, and time polynomial in $1/\gamma$,$1/\epsilon_1$ and $\log(1/\delta)$.

3.3 Extensions

Combining Multiple Similarity Functions: Suppose that rather than having a single similarity function, we were instead given n functions $K_1, \ldots, K_n$, and our hope is that some convex combination of them will satisfy Definition 8. 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 3 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 [15] 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 5. *Suppose* $K_1, \ldots, K_n$ *are similarity functions such that some (unknown) convex combination of them is* (ϵ,γ)*-good. If one draws a set* $S = \{\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_d\}$ *from* P *containing* $d = 8\log(1/\delta)/\gamma^2$ *examples, then with probability at least* $1-\delta$*, the mapping* $\phi^S : X \to R^{nd}$ *defined as* $\phi^S(x) = \frac{\rho^S(x)}{\sqrt{nd}}$*,*

$$\rho^S(x) = (K_1(x,\tilde{x}_1), \ldots, K_1(x,\tilde{x}_d), ..., K_n(x,\tilde{x}_1), \ldots, K_n(x,y_d))$$

has the property that the induced distribution $\phi^S(P)$ *in* R^{nd} *has a separator of error at most* $\epsilon + \delta$ *at margin at least* $\gamma/(2\sqrt{n})$*.*

Proof. Let $K = \alpha_1 K_1 + \ldots + \alpha_n K_n$ be an (ϵ, γ)-good convex-combination of the K_i. By Theorem 3, had we instead performed the mapping: $\hat{\phi}^S : X \to R^d$ defined as $\hat{\phi}^S(x) = \frac{\hat{\rho}^S(x)}{\sqrt{d}}$,

$$\hat{\rho}^S(x) = (K(x, \tilde{x}_1), \ldots, K(x, \tilde{x}_d))$$

then with probability $1 - \delta$, the induced distribution $\hat{\phi}^S(P)$ in R^d would have a separator of error at most $\epsilon + \delta$ at margin at least $\gamma/2$. Let $\hat{\beta}$ be the vector corresponding to such a separator in that space. Now, let us convert $\hat{\beta}$ into a vector in R^{nd} by replacing each coordinate $\hat{\beta}_j$ with the n values $(\alpha_1 \hat{\beta}_j, \ldots, \alpha_n \hat{\beta}_j)$. Call the resulting vector $\tilde{\beta}$. Notice that by design, for any x we have $\left\langle \tilde{\beta}, \phi^S(x) \right\rangle = \frac{1}{\sqrt{n}} \left\langle \hat{\beta}, \hat{\phi}^S(x) \right\rangle$. Furthermore, $\left\| \tilde{\beta} \right\| \leq \left\| \hat{\beta} \right\| \leq 1$ (the worst case is when exactly one of the α_i is equal to 1 and the rest are 0). Thus, the vector $\tilde{\beta}$ under distribution $\phi^S(P)$ has the similar properties as the vector $\hat{\beta}$ under $\hat{\phi}^S(P)$; so, using the proof of Theorem 3 we obtain that that the induced distribution $\phi^S(P)$ in R^{nd} has a separator of error at most $\epsilon + \delta$ at margin at least $\gamma/(2\sqrt{n})$. □

Note that the above argument actually shows something a bit stronger than Theorem 5. In particular, if we define $\alpha = (\alpha_1, \ldots, \alpha_n)$ to be the mixture vector for the optimal K, then we can replace the margin bound $\gamma/(2\sqrt{n})$ with $\gamma/(2\left\|\alpha\right\|\sqrt{n})$. For example, if α is the uniform mixture, then we just get the bound in Theorem 3 of $\gamma/2$.

Multi-class Classification: We can naturally extend all our results to multi-class classification. Assume for concreteness that there are r possible labels, and denote the space of possible labels by $Y = \{1, \cdots, r\}$; thus, by a *multi-class learning problem* we mean a distribution P over labeled examples $(x, y(x))$, where $x \in X$ and $y(x) \in Y$.

For this multi-class setting, Definition 7 seems most natural to extend. Specifically:

Definition 10 (main, multi-class). *A similarity function K is an (ϵ, γ)-***good similarity function** *for a multi-class 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:*

$$\mathbf{E}_{x' \sim P}[w(x')K(x, x')|y(x) = y(x')] \geq \mathbf{E}_{x' \sim P}[w(x')K(x, x')|y(x) = i] + \gamma \text{ for all } i \in Y, i \neq y(x)$$

We can then extend the argument in Theorem 3 and learn using standard adaptations of linear-separator algorithms to the multiclass case (e.g., see [8]).

4 Relationship Between Kernels and Similarity Measures

As discussed earlier, the similarity-based theory of learning is more general than the traditional kernel-based theory, since a good similarity function need not be a valid kernel. However, for a similarity function K that is a valid kernel, it is interesting to understand the relationship between the learning results guaranteed by the two theories. Similar learning guarantees and sample complexity bounds can be obtained if K is either an (ϵ, γ)-good similarity function, or a valid kernel and (ϵ, γ)-kernel-good. In fact, as we saw in Section 3.2, the similarity-based guarantees are obtained by transforming (using a sample) the problem of learning with an (ϵ, γ)-good similarity function to learning with a kernel with essentially the same goodness parameters. This is made more explicit in Section 4.1. Understanding the relationship between the learning guarantees then boils down to understanding the relationship between the two notions of goodness.

In this Section we study the relationship between a kernel function being good in the similarity sense and good in the kernel sense. We show that a valid kernel function that is good for one notion, is in fact good also for the other notion. The qualitative notions of being "good" are therefore equivalent for valid kernels, and so in this sense the more general similarity-based notion subsumes the familiar kernel-based notion.

However, as we will see, the similarity-based margin of a valid kernel might be lower than the kernel-based margin, yielding a possible increase in the sample complexity guarantees if a kernel is used as a similarity measure. Since we will show that for a valid kernel, the kernel-based margin is never smaller than the similarity-based margin, we see that the similarity-based notion, despite being more general, is strictly less powerful quantitatively on those similarity functions for which the kernel theory applies. We provide a tight bound on this possible deterioration of the margin when switching to the similarity-based notion.

Specifically, we show that if a valid kernel function is good in the similarity sense, it is also good in the standard kernel sense, both for the margin violation error rate and for the hinge loss:

Theorem 6 (A kernel good as a similarity function is also good as a kernel). *If K is a valid kernel function, and is (ϵ, γ)-good similarity for some learning problem, then it is also (ϵ, γ)-kernel-good for the learning problem. If K is (ϵ, γ)-good similarity in hinge loss, then it is also (ϵ, γ)-kernel-good in hinge loss.*

We also show the converse—If a kernel function is good in the kernel sense, it is also good in the similarity sense, though with some degradation of the margin:

Theorem 7 (A good kernel is also a good similarity function—Margin violations). *If K is (ϵ_0, γ)-kernel-good for some learning problem (with deterministic labels), then it is also $(\epsilon_0 + \epsilon_1, \frac{1}{2}(1 - \epsilon_0)\epsilon_1\gamma^2)$-good similarity for the learning problem, for any $\epsilon_1 > 0$.*

Note that in any useful situation $\epsilon_0 < \frac{1}{2}$, and so the guaranteed margin is at least $\frac{1}{4}\epsilon_1\gamma^2$. A similar guarantee holds also for the hinge loss:

Theorem 8 (A good kernel is also a good similarity function—Hinge loss). *If K is (ϵ_0, γ)-kernel-good in hinge loss for learning problem (with deterministic labels), then it is also $(\epsilon_0 + \epsilon_1, 2\epsilon_1\gamma^2)$-good similarity in hinge loss for the learning problem, for any $\epsilon_1 > 0$.*

These results establish that treating a kernel as a similarity function would still enable learning, although with a somewhat increased sample complexity. Unfortunately, the deterioration of the margin in the above results, which yields an increase in the sample complexity guarantees, is unavoidable:

Theorem 9 (Tightness, Margin Violations). *For any $0 < \gamma < \sqrt{\frac{1}{2}}$ and any $0 < \epsilon_1 < \frac{1}{2}$, there exists a learning problem and a kernel function K, which is $(0, \gamma)$-kernel-good for the learning problem, but which is only $(\epsilon_1, 4\epsilon_1\gamma^2)$-good similarity. That is, it is not (ϵ_1, γ')-good similarity for any $\gamma' > 4\epsilon_1\gamma^2$.*

Theorem 10 (Tightness, Hinge Loss). *For any $0 < \gamma < \sqrt{\frac{1}{2}}$ and any $0 < \epsilon_1 < \frac{1}{2}$, there exists a learning problem and a kernel function K, which is $(0, \gamma)$-kernel-good in hinge loss for the learning problem, but which is only $(\epsilon_1, 32\epsilon_1\gamma^2)$-good similarity in hinge loss.*

To prove Theorem 6 we will show, for any weight function, an explicit low-norm linear predictor β (in the implied Hilbert space), with equivalent behavior (Section 4.2). To prove Theorems 7 and 8, we will consider a kernel function that is (ϵ_0, γ)-kernel-good and show that it is also good as a similarity function. We will first treat goodness in hinge-loss and prove Theorem 8 (Section 4.3), which can be viewed as a more general result. This will be done using the representation of the optimal SVM solution in terms of the dual optimal solution. Then, in Section 4.4, we prove Theorem 7 in terms of the margin violation error rate, by using the hinge-loss as a bound on the error rate. To prove Theorems 9 and 10, we present an explicit learning problem and kernel (Section 4.5).

4.1 Transforming a Good Similarity Function to a Good Kernel

Before proving the above Theorems, we briefly return to the mapping of Theorem 3 and explicitly present it as a mapping between a good similarity function and a good kernel:

Corollary 1 (A good similarity function can be transformed to a good kernel). *If K is an (ϵ, γ)-good similarity function for some learning problem P, then for any $0 < \delta < 1$, given a sample of S size $(8/\gamma^2)\log(1/\delta)$ drawn from P, we can construct, with probability at least $1 - \delta$ over the draw of S, a valid kernel $\tilde{K}^S$ that is $(\epsilon + \delta, \gamma/2)$-kernel good for P.*

If K is a (ϵ, γ)-good similarity function in hinge-loss for some learning problem P, then for any $\epsilon_1 > 0$ and $0 < \delta < \gamma\epsilon_1/4$, given a sample of S size $16\log(1/\delta)/(\epsilon_1\gamma)^2$ drawn from P, we can construct, with probability at least $1 - \delta$ over the draw of S, a valid kernel $\tilde{K}^S$ that is $(\epsilon + \epsilon_1, \gamma)$-kernel good for P.

Proof. Let $\tilde{K}^S(x,x') = \left\langle \phi^S(x), \phi^S(x') \right\rangle$ where ϕ^S is the transformation of Theorems 3 and 4.

From this statement, it is clear that kernel-based learning guarantees apply also to learning with a good similarity function, essentially with the same parameters.

It is important to understand that the result of Corollary 1 is of a very different nature than the results of Theorems 6– 10. The claim here is not that a good similarity function *is* a good kernel — it can't be if it is not positives semi-definite. But, given a good similarity function we can create a good kernel. This transformation is *distribution-dependent*, and can be calculated using a sample S.

4.2 Proof of Theorem 6

Consider a similarity function K that is a valid kernel, i.e. $K(x,x') = \langle \phi(x), \phi(x') \rangle$ for some mapping ϕ of x to a Hilbert space $\mathcal{H}$. For any input distribution and any valid weighting $w(x)$ of the inputs (i.e. $0 \leq w(x) \leq 1$), we will construct a linear predictor $\beta_w \in \mathcal{H}$, with $\|\beta_w\| \leq 1$, such that similarity-based predictions using w are the same as the linear predictions made with β_w

Define the following linear predictor $\beta_w \in \mathcal{H}$:

$$\beta_w = \mathbf{E}_{x'}[y(x')w(x')\phi(x')]. \tag{4.1}$$

The predictor β_w has norm at most:

$$\|\beta_w\| = \|\mathbf{E}_{x'}[y(x')w(x')\phi(x')]\| \leq \max_{x'} \|y(x')w(x')\phi(x')\| \leq \max \|\phi(x')\| = \max \sqrt{K(x',x')} \leq 1 \tag{4.2}$$

where the second inequality follows from $|w(x')|, |y(x')| \leq 1$.

The predictions made by β_w are:

$$\langle \beta_w, \phi(x) \rangle = \langle \mathbf{E}_{x'}[y(x')w(x')\phi(x')], \phi(x) \rangle = \mathbf{E}_{x'}[y(x')w(x')\langle \phi(x'), \phi(x) \rangle] = \mathbf{E}_{x'}[y(x')w(x')K(x,x')] \tag{4.3}$$

That is, using β_w is the same as using similarity-based prediction with w. In particular, if the margin violation rate, as well as the hinge loss, with respect to any margin γ, is the same for predictions made using either w or β_w. This is enough to establish Theorem 6: If K is (ϵ, γ)-good (perhaps for to the hinge-loss), there exists some valid weighting w the yields margin violation error rate (resp. hinge loss) at most ϵ with respect to margin γ, and so β_w yields the same margin violation (resp. hinge loss) with respect to the same margin, establishing K is (ϵ, γ)-kernel-good (resp. for the hinge loss).

4.3 Proof of Theorem 8: Guarantee on the Hinge Loss

Recall that we are considering only learning problems where the label y is a deterministic function of x. For simplicity of presentation, we first consider finite discrete distributions, where:

$$\Pr(\, x_i, y_i \,) = p_i \tag{4.4}$$

for $i = 1 \ldots n$, with $\sum_{i=1}^n p_i = 1$ and $x_i \neq x_j$ for $i \neq j$.

Let K be any kernel function that is (ϵ_0, γ)-kernel good in hinge loss.Let ϕ be the implied feature mapping and denote $\phi_i = \phi(x_i)$. Consider the following weighted-SVM quadratic optimization problem with regularization parameter C:

$$\text{minimize } \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^n p_i[1 - y_i\langle \beta, \phi_i \rangle]_+ \tag{4.5}$$

The dual of this problem, with dual variables α_i, is:

$$\begin{aligned} \text{maximize } & \sum_i \alpha_i - \frac{1}{2}\sum_{ij} y_i y_j \alpha_i \alpha_j K(x_i, x_j) \\ \text{subject to } & \; 0 \leq \alpha_i \leq C p_i \end{aligned} \tag{4.6}$$

There is no duality gap, and furthermore the primal optimum β^* can be expressed in terms of the dual optimum α^*: $\beta^* = \sum_i \alpha_i^* y_i x_i$.

Since K is (ϵ_0, γ)-kernel-good in hinge-loss, there exists a predictor $\|\beta_0\| = 1$ with average-hinge loss ϵ_0 relative to margin γ. The primal optimum β^* of (4.5), being the optimum solution, then satisfies:

$$\begin{aligned}\frac{1}{2}\|\beta^*\|^2 + C\sum_i p_i[1 - y_i\langle\beta^*, \phi_i\rangle]_+ &\leq \frac{1}{2}\left\|\frac{1}{\gamma}\beta_0\right\|^2 + C\sum_i p_i[1 - y_i\left\langle\frac{1}{\gamma}\beta_0, \phi_i\right\rangle]_+ \\ &= \frac{1}{2\gamma^2} + C\mathbf{E}\left[[1 - y\left\langle\frac{1}{\gamma}\beta_0, \phi(x)\right\rangle]_+\right] = \frac{1}{2\gamma^2} + C\epsilon_0 \qquad (4.7)\end{aligned}$$

Since both terms on the left hand side are non-negative, each of them is bounded by the right hand side, and in particular:

$$C\sum_i p_i[1 - y_i\langle\beta^*, \phi_i\rangle]_+ \leq \frac{1}{2\gamma^2} + C\epsilon_0 \qquad (4.8)$$

Dividing by C we get a bound on the average hinge-loss of the predictor β^*, relative to a margin of one:

$$\mathbf{E}[[1 - y\langle\beta^*, \phi(x)\rangle]_+] \leq \frac{1}{2C\gamma^2} + \epsilon_0 \qquad (4.9)$$

We now use the fact that β^* can be written as $\beta^* = \sum_i \alpha_i^* y_i \phi_i$ with $0 \leq \alpha_i^* \leq Cp_i$. Using the weights

$$w_i = w(x_i) = \alpha_i^*/(Cp_i) \leq 1 \qquad (4.10)$$

we have for every x, y:

$$\begin{aligned}y\mathbf{E}_{x',y'}[w(x')y'K(x,x')] &= y\sum_i p_i w(x_i) y_i K(x, x_i) \qquad (4.11)\\ &= y\sum_i p_i \alpha_i^* y_i K(x, x_i)/(Cp_i) \\ &= y\sum_i \alpha_i^* y_i \langle\phi_i, \phi(x)\rangle/C = y\langle\beta^*, \phi(x)\rangle/C\end{aligned}$$

Multiplying by C and using (4.9):

$$\mathbf{E}_{x,y}[\,[\,1 - Cy\mathbf{E}_{x',y'}[w(x')y'K(x,x')]\,]_+\,] = \mathbf{E}_{x,y}[\,[\,1 - y\langle\beta^*, \phi(x)\rangle\,]_+\,] \leq \frac{1}{2C\gamma^2} + \epsilon_0 \qquad (4.12)$$

This holds for any C, and describes the average hinge-loss relative to margin $1/C$. To get an average hinge-loss of $\epsilon_0 + \epsilon_1$, we set $C = 1/(2\epsilon_1\gamma^2)$ and get:

$$\mathbf{E}_{x,y}\left[\,[\,1 - y\mathbf{E}_{x',y'}[w(x')y'K(x,x')]/(2\epsilon_1\gamma^2)\,]_+\,\right] \leq \epsilon_0 + \epsilon_1 \qquad (4.13)$$

This establishes that K is $(\epsilon_0 + \epsilon_1, 2\epsilon_1\gamma^2)$-good similarity in hinge-loss.

Non-discrete distributions The same arguments apply also in the general (not necessarily discrete) case, except that this time, instead of a fairly standard (weighted) SVM problem, we must deal with a variational optimization problem, where the optimization variable is a random variable (a function from the sample space to the reals). We will present the dualization in detail.

We consider the primal objective

$$\text{minimize } \frac{1}{2}\|\beta\|^2 + C\mathbf{E}_{y,\phi}[[1 - y\langle\beta, \phi\rangle]_+] \qquad (4.14)$$

where the expectation is w.r.t. the distribution P, with $\phi = \phi(x)$ here and throughout the rest of this section. We will rewrite this objective using explicit slack, in the form of a random variable ξ, which will be a variational optimization variable:

$$\begin{aligned} &\text{minimize } \frac{1}{2}\|\beta\|^2 + C\mathbf{E}[\xi] \\ &\text{subject to } \Pr(\, 1 - y\langle\beta, \phi\rangle - \xi \le 0 \,) = 1 \\ &\qquad\qquad\; \Pr(\, \xi \ge 0 \,) = 1 \end{aligned} \tag{4.15}$$

In the rest of this section all our constraints will implicitly be required to hold with probability one. We will now introduce the dual variational optimization variable α, also a random variable over the same sample space, and write the problem as a saddle problem:

$$\begin{aligned} &\min_{\beta,\xi} \max_{\alpha} \frac{1}{2}\|\beta\|^2 + C\mathbf{E}[\xi] + \mathbf{E}[\alpha(1 - y\langle\beta, \phi\rangle - \xi)] \\ &\text{subject to} \quad \xi \ge 0 \quad \alpha \ge 0 \end{aligned} \tag{4.16}$$

Note that this choice of Lagrangian is a bit different than the more standard Lagrangian leading to (4.6). Convexity and the existence of a feasible point in the dual interior allows us to change the order of maximization and minimization without changing the value of the problem, even in the infinite case [10]. Rearranging terms we obtaining the equivalent problem:

$$\begin{aligned} &\max_{\alpha} \min_{\beta,\xi} \frac{1}{2}\|\beta\|^2 - \langle\mathbf{E}[\alpha y\phi], \beta\rangle + \mathbf{E}[\xi(C - \alpha)] + \mathbf{E}[\alpha] \\ &\text{subject to} \quad \xi \ge 0, \quad \alpha \ge 0 \end{aligned} \tag{4.17}$$

Similarly to the finite case, we see that the minimum of the minimization problem is obtained when $\beta = \mathbf{E}[\alpha y\phi]$ and that it is finite when $\alpha \le C$ almost surely, yielding the dual:

$$\begin{aligned} &\text{maximize } \mathbf{E}[\alpha] - \frac{1}{2}\mathbf{E}[\alpha y\alpha' y K(x, x')] \\ &\text{subject to} \quad 0 \le \alpha \le C \end{aligned} \tag{4.18}$$

where (x, y, α) and (x', y', α') are two independent draws from the same distribution. The primal optimum can be expressed as $\beta^* = \mathbf{E}[\alpha^* y\phi]$, where α^* is the dual optimum. We can now apply the same arguments as in (4.7),(4.8) to get (4.9). Using the weight mapping

$$w(x) = \mathbf{E}[\alpha^*|x] \,/\, C \le 1 \tag{4.19}$$

we have for every x, y:

$$y\mathbf{E}_{x',y'}[w(x')y'K(x, x')] = y\langle\mathbf{E}_{x',y',\alpha'}[\alpha' y' x'], x\rangle/C = y\langle\beta^*, \phi(x)\rangle/C. \tag{4.20}$$

From here we can already get (4.12) and setting $C = 1/(2\epsilon_1\gamma^2)$ we get (4.13), which establishes Theorem 8 for any learning problem (with deterministic labels).

4.4 Proof of Theorem 7: Guarantee on Margin Violations

We will now turn to guarantees on similarity-goodness with respect to the margin violation error-rate. We base these on the results for goodness in hinge loss, using the hinge loss as a bound on the margin violation error-rate. In particular, a violation of margin $\gamma/2$ implies a hinge-loss at margin γ of at least $\frac{1}{2}$. Therefore, twice the average hinge-loss at margin γ is an upper bound on the margin violation error rate at margin $\gamma/2$.

The kernel-separable case, i.e. $\epsilon_0 = 0$, is simpler, and we consider it first. Having no margin violations implies zero hinge loss. And so if a kernel K is $(0, \gamma)$-kernel-good, it is also $(0, \gamma)$-kernel-good in hinge loss, and by Theorem 8 it is $(\epsilon_1/2, 2(\epsilon_1/2)\gamma^2)$-good similarity in hinge loss. Now, for any $\epsilon_1 > 0$, by bounding the margin $\frac{1}{2}\epsilon_1\gamma^2$ error-rate by the $\epsilon_1\gamma^2$ average hinge loss, K is $(\epsilon_1, \frac{1}{2}\epsilon_1\gamma^2)$-good similarity, establishing Theorem 7 for the case $\epsilon_0 = 0$.

We now return to the non-separable case, and consider a kernel K that is (ϵ_0, γ)-kernel-good, with some non-zero error-rate ϵ_0. Since we cannot bound the hinge loss in terms of the margin-violations, we will instead consider a modified distribution where the margin-violations are removed.

Let β^* be the linear classifier achieving ϵ_0 margin violation error-rate with respect to margin γ, i.e. such that $\Pr(\, y\langle \beta^*, x\rangle \geq \gamma\,) > 1 - \epsilon_0$. We will consider a distribution which is conditioned on $y\langle \beta^*, x\rangle \geq \gamma$. We denote this event as $\text{OK}(x)$ (recall that y is a deterministic function of x). The kernel K is obviously $(0, \gamma)$-kernel-good, and so by the arguments above also $(\epsilon_1, \frac{1}{2}\epsilon_1\gamma^2)$-good similarity, on the conditional distribution. Let w be the weight mapping achieving

$$\Pr_{x,y}(\, y\mathbf{E}_{x',y'}[w(x')y'K(x,x')|\text{OK}(x')] < \gamma_1 | \text{OK}(x)\,) \leq \epsilon_1, \tag{4.21}$$

where $\gamma_1 = \frac{1}{2}\epsilon_1\gamma^2$, and set $w(x) = 0$ when $\text{OK}(x)$ does not hold. We have:

$$\begin{aligned}
&\Pr_{x,y}(\, y\mathbf{E}_{x',y'}[w(x')y'K(x,x')] < (1-\epsilon_0)\gamma_1\,) \\
&\qquad \leq \Pr(\,\text{not } \text{OK}(x)\,) + \Pr(\,\text{OK}(x)\,)\Pr_{x,y}(\;\, y\mathbf{E}_{x',y'}[w(x')y'K(x,x')] < (1-\epsilon_0)\gamma_1 \mid \text{OK}(x)\;\,) \\
&\qquad = \epsilon_0 + (1-\epsilon_0)\Pr_{x,y}(\, y(1-\epsilon_0)\mathbf{E}_{x',y'}[w(x')y'K(x,x')|\text{OK}(x)] < (1-\epsilon_0)\gamma_1|\text{OK}(x)\,) \\
&\qquad = \epsilon_0 + (1-\epsilon_0)\Pr_{x,y}(\, y\mathbf{E}_{x',y'}[w(x')y'K(x,x')|\text{OK}(x)] < \gamma_1|\text{OK}(x)\,) \\
&\qquad \leq \epsilon_0 + (1-\epsilon_0)\epsilon_1 \leq \epsilon_0 + \epsilon_1
\end{aligned} \tag{4.22}$$

establishing that K is $(\epsilon_0 + \epsilon_1, \gamma_1)$-good similarity for the original (unconditioned) distribution, thus yielding Theorem 7.

4.5 Tightness

We now turn to proving of Theorems 9 and 10. This is done by presenting a specific distribution P and kernel in which the guarantees hold tightly.

Consider the standard Euclidean inner-product and a distribution on four labeled points in $\mathbb{R}^3$, given by:

$$\begin{aligned}
x_1 &= (\gamma, \gamma, \sqrt{1-2\gamma^2}), & y_1 &= 1, & p_1 &= \frac{1}{2} - \epsilon \\
x_2 &= (\gamma, -\gamma, \sqrt{1-2\gamma^2}), & y_2 &= 1, & p_2 &= \epsilon \\
x_3 &= (-\gamma, \gamma, \sqrt{1-2\gamma^2}), & y_3 &= -1, & p_3 &= \epsilon \\
x_4 &= (-\gamma, -\gamma, \sqrt{1-2\gamma^2}), & y_4 &= -1, & p_4 &= \frac{1}{2} - \epsilon
\end{aligned}$$

for some (small) $0 < \gamma < \sqrt{\frac{1}{2}}$ and (small) probability $0 < \epsilon < \frac{1}{2}$. The four points are all on the unit sphere (i.e. $\|x_i\| = 1$ and so $K(x_i, x_j) = \langle x_i, x_j\rangle \leq 1$), and are clearly separated by $\beta = (1, 0, 0)$ with a margin of γ. The standard inner-product kernel is therefore $(0, \gamma)$-kernel-good on this distribution.

Proof of Theorem 9: Tightness for Margin-Violations We will show that when this kernel (the standard inner product kernel in $\mathbb{R}^3$) is used as a similarity function, the best margin that can be obtained on all four points, i.e. on at least $1 - \epsilon$ probability mass of examples, is $8\epsilon\gamma^2$.

Consider the classification margin on point x_2 with weights w (denote $w_i = w(x_i)$):

$$\begin{aligned}
&\mathbf{E}[w(x)yK(x_2, x)] \\
&= (\frac{1}{2} - \epsilon)w_1(\gamma^2 - \gamma^2 + (1-2\gamma^2)) + \epsilon w_2(2\gamma^2 + (1-2\gamma^2)) \\
&\quad -\epsilon w_3(-2\gamma^2 + (1-2\gamma^2)) - (\frac{1}{2} - \epsilon)w_4(-\gamma^2 + \gamma^2 + (1-2\gamma^2)) \\
&= \left((\frac{1}{2} - \epsilon)(w_1 - w_4) + \epsilon(w_2 - w_3)\right)(1-2\gamma^2) \,+\, 2\epsilon(w_2 + w_3)\gamma^2
\end{aligned} \tag{4.23}$$

If the first term is positive, we can consider the symmetric calculation

$$-\mathbf{E}[w(x)yK(x_3,x)] = -\left((\frac{1}{2}-\epsilon)(w_1-w_4)+\epsilon(w_2-w_3)\right)(1-2\gamma^2)\ +\ 2\epsilon(w_2+w_3)\gamma^2$$

in which the first term is negated. One of the above margins must therefore be at most

$$2\epsilon(w_2+w_3)\gamma^2 \leq 4\epsilon\gamma^2 \tag{4.24}$$

This establishes Theorem 9.

4.6 Proof of Theorem 10: Tightness for the Hinge Loss

In the above example, suppose we would like to get an average hinge-loss relative to margin γ_1 of at most ϵ_1:

$$\mathbf{E}_{x,y}[\,[\,1-y\mathbf{E}_{x',y'}[w(x')y'K(x,x')]/\gamma_1\,]_+\,] \leq \epsilon_1 \tag{4.25}$$

Following the arguments above, equation (4.24) can be used to bound the hinge-loss on at least one of the points x_2 or x_3, which, multiplied by the probability ϵ of the point, is a bound on the average hinge loss:

$$\mathbf{E}_{x,y}[\,[\,1-y\mathbf{E}_{x',y'}[w(x')y'K(x,x')]/\gamma_1\,]_+\,] \geq \epsilon(1-4\epsilon\gamma^2/\gamma_1) \tag{4.26}$$

and so to get an an average hinge-loss of at most ϵ_1 we must have:

$$\gamma_1 \leq \frac{4\epsilon\gamma^2}{1-\epsilon_1/\epsilon} \tag{4.27}$$

For any target hinge-loss ϵ_1, consider a distribution with $\epsilon = 2\epsilon_1$, in which case we get that the maximum margin attaining average hinge-loss ϵ_1 is $\gamma_1 = 16\epsilon_1\gamma^2$, even though we can get a hinge loss of zero at margin γ using a kernel. This establishes Theorem 10.

Note: One might object that the example used in Theorems 9 and 10 is a bit artificial, since K has margin $O(\gamma^2)$ in the similarity sense just because $1-4\gamma^2 \leq K(x_i,x_j) \leq 1$. Normalizing K to $[-1,1]$ we would obtain a similarity function that has margin $O(1)$. However, this "problem" can be simply fixed by adding the symmetric points on the lower semi-sphere:

$$\begin{array}{lll}
x_5 = (\gamma,\gamma,-\sqrt{1-2\gamma^2}), & y_5 = 1, & p_5 = \dfrac{1}{4}-\epsilon \\
x_6 = (\gamma,-\gamma,-\sqrt{1-2\gamma^2}), & y_6 = 1, & p_6 = \epsilon \\
x_7 = (-\gamma,\gamma,-\sqrt{1-2\gamma^2}), & y_7 = -1, & p_7 = \epsilon \\
x_8 = (-\gamma,-\gamma,-\sqrt{1-2\gamma^2}), & y_8 = -1, & p_8 = \dfrac{1}{4}-\epsilon
\end{array}$$

and by changing $p_1 = \frac{1}{4}-\epsilon$ and $p_4 = \frac{1}{4}-\epsilon$. The classification margins on x_2 and x_3 are now (compare with (4.23)):

$$\begin{aligned}
\mathbf{E}[w(x)yK(x_2,x)] &= \left((\frac{1}{4}-\epsilon)(w_1-w_4-w_5+w_8)+\epsilon(w_2-w_3-w_6+w_7)\right)(1-2\gamma^2) \\
&\quad + 2\epsilon(w_2+w_3+w_6+w_7)\gamma^2 \\
-\mathbf{E}[w(x)yK(x_3,x)] &= -\left((\frac{1}{4}-\epsilon)(w_1-w_4-w_5+w_8)+\epsilon(w_2-w_3-w_6+w_7)\right)(1-2\gamma^2) \\
&\quad + 2\epsilon(w_2+w_3+w_6+w_7)\gamma^2
\end{aligned}$$

One of the above classification margins must therefore be at most $2\epsilon(w_2+w_3+w_6+w_7)\gamma^2 \leq 8\epsilon\gamma^2$. Thus, even though the similarity is "normalized" and $(0,\gamma)$-kernel-good, it is only $(\epsilon, 8\epsilon\gamma^2)$-good as a similarity function. Proceeding as in the proof of Theorem 10 establishes the modified example is also only $(\epsilon, 64\epsilon\gamma^2)$-good in hinge loss.[6]

[6] We thank the anonymous referee for suggesting this strengthening of the lower bound.

5 Probabilistic Labels

So far, we have considered only learning problems where the label y is a deterministic function of x. Here, we discuss the necessary modifications to extend our theory also to noisy learning problems, where the same point x might be associated with both positive and negative labels with positive probabilities.

Although the learning guarantees of Section 3 are valid also for noisy learning problems, a kernel that is kernel-good for a noisy learning problem might not be good as a similarity function for this learning problem. To amend this, the definition of a good similarity function must be corrected, allowing the weights to depend not only on the point x but also on the label y:

Definition 11 (Main, Margin Violations, Corrected for Noisy Problems). *A similarity function K is an (ϵ, γ)-***good similarity function** *for a learning problem P if there* exists *a bounded weighting function w over $X \times \{-1, +1\}$ ($w(x', y') \in [0, 1]$ for all $x' \in X, y' \in \{-1, +1\}$) such that at least a $1 - \epsilon$ probability mass of examples x, y satisfy:*

$$\mathbf{E}_{x',y' \sim P}[yy'w(x', y')K(x, x')] \geq \gamma. \tag{5.1}$$

It is easy to verify that Theorem 3 can be extended also to this corrected definition. The same mapping ϕ^S can be used, with $\beta_i = \tilde{y}_i w(\tilde{x}_i, \tilde{y}_i)$, where $\tilde{y}_i$ is the training label of example i. Definition 9 and Theorem 4 can be extended in a similar way.

With these modified definitions, Theorems 7 and 8 extend also to noisy learning problems. In the proof of Theorem 8, two of the points x_i, x_j might be identical, but have different labels $y_i = 1, y_j = -1$ associated with them. This might lead to two different weights w_i, w_j for the same point. But since w is now allowed to depend also on the label, this does not pose a problem. In the non-discrete case, this corresponds to defining the weight as:

$$w(x, y) = \mathbf{E}[\alpha^* | x, y] \,/\, C. \tag{5.2}$$

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 semi-definite. 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.

It would be interesting to analyze alternative sufficient conditions for learning via pairwise functions. Although in this work we established guarantees for learning with a good similarity function by transforming the problem to learning a linear separator, we would like to emphasize that this transformation was used as a convenient tool. For other notions of "goodness" of pairwise functions, it might well be more convenient to establish learnability without reference to linear separation.

From a practical perspective, the results of Section 4 suggest that if K is in fact a valid kernel, we are probably better off using it as a kernel, e.g. in an SVM or Perceptron algorithm, rather than going through the transformation of Section 3.2. However, faced with a non-positive-semidefinite similarity function (coming from domain experts), the transformation of Theorem 3 might well be useful. In fact, Liao and Noble have used an algorithm similar to the one we propose in the context of protein classification [16]. Furthermore, a direct implication of Theorem 6 is that we can indeed think (in the design process) of the usefulness of a kernel function in terms of more intuitive, direct properties of the data in the original representation, without need to refer to implicit spaces.

Finally, our algorithms (much like those of [5]) 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 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.

Subsequent Work: Inspired by our work, Wang et. al [28] have recently analyzed different, alternative sufficient conditions for learning via pairwise functions. In particular, Wang et. al [28] analyze unbounded dissimilarity functions which are invariant to order preserving transformations. They provide conditions that they prove are sufficient for learning, though they may not include all good kernel functions.

On a different line of inquiry, Balcan et. al [6] use our approach for analyzing similarity functions in the context of *clustering* (i.e. learning from purely *unlabeled* data). Specifically, Balcan et. al [6] asks what (stronger) properties would be sufficient to allow one to produce an accurate hypothesis without any label information at all. Balcan et. al [6] show that if one relaxes the objective (for example, allows the algorithm to produce a hierarchical clustering such that some pruning is close to the correct answer), then one can define a number of interesting graph-theoretic and game-theoretic properties of similarity functions that are sufficient to cluster well.

Acknowledgments: We would like to thank the anonymous referees for their insightful and helpful comments. This work was supported in part by the National Science Foundation under grants CCF-0514922, CCR-0122581, and IIS-0121678, as well as by a Google Research Grant and an IBM Graduate Fellowship. Part of this work was done while NS was visiting IBM Haifa Research Labs.

References

1. *http://www.kernel-machines.org/*.
2. M. Anthony and P. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 1999.
3. S. Arora, L. Babai, J. Stern, and Z. Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. *Journal of Computer and System Sciences*, 54:317 – 331, 1997.
4. M.-F. Balcan and A. Blum. On a theory of learning with similarity functions. In *International Conference on Machine Learning*, 2006.
5. M.-F. Balcan, A. Blum, and S. Vempala. Kernels as features: On kernels, margins, and low-dimensional mappings. *Machine Learning*, 65(1):79 – 94, 2006.
6. M.-F. Balcan, A. Blum, and S. Vempala. A discriminative framework for clustering via similarity functions. In *Proceedings of the 40th ACM Symposium on Theory of Computing*, 2008.
7. P. L. Bartlett and S. Mendelson. Rademacher and gaussian complexities: risk bounds and structural results. *J. Mach. Learn. Res.*, 3:463–482, 2003.
8. Y. Freund and R. E. Schapire. Large margin classification using the perceptron algorithm. *Machine Learning*, 37(3):277 – 296, 1999.
9. R. Herbrich. *Learning Kernel Classifiers*. MIT Press, Cambridge, 2002.
10. R. Hettich and K. O. Kortanek. Semi-infinite programming: theory, methods, and applications. *SIAM Rev.*, 35(3):380–429, 1993.
11. T. S. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. In *Advances in neural information processing systems 11*. MIT Press, 1999.
12. T. Joachims. *Learning to Classify Text Using Support Vector Machines: Methods, Theory, and Algorithms*. Kluwer, 2002.
13. A. Kalai, A. Klivans, Y. Mansour, and R. Servedio. Agnostically learning halfspaces. In *Proceedings of the 46th Annual Symposium on the Foundations of Computer Science*, 2005.
14. M. Kearns and U. Vazirani. *An Introduction to Computational Learning Theory*. MIT Press, 1994.
15. G. R. G. Lanckriet, N. Cristianini, P. L. Bartlett, L. El Ghaoui, and M. I. Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 5:27–72, 2004.
16. L. Liao and W. S. Noble. Combining pairwise sequence similarity and support vector machines for detecting remote protein evolutionary and structural relationships. *Journal of Computational Biology*, 10(6):857–868, 2003.
17. D. McAllester. Simplified pac-bayesian margin bounds. In *Proceedings of the 16th Conference on Computational Learning Theory*, 2003.
18. R.E. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.
19. B. Scholkopf and A. J. Smola. *Learning with kernels. Support Vector Machines, Regularization, Optimization, and Beyond*. MIT University Press, Cambridge, 2002.

20. B. Scholkopf, K. Tsuda, and J.-P. Vert. *Kernel Methods in Computational Biology.* MIT Press, 2004.
21. J. Shawe-Taylor, P. L. Bartlett, R. C. Williamson, and M. Anthony. Structural risk minimization over data-dependent hierarchies. *IEEE Transactions on Information Theory*, 44(5):1926–1940, 1998.
22. J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis.* Cambridge University Press, 2004.
23. A. J. Smola and B. Schölkopf. *Learning with Kernels.* MIT Press, 2002.
24. N. Srebro. How Good is a Kernel as a Similarity Function. *COLT* , 2007.
25. L.G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, 1984.
26. V. N. Vapnik. *Statistical Learning Theory*. John Wiley and Sons Inc., 1998.
27. S.V.N. Viswanathan and A. J. Smola. Fast kernels for string and tree matching. In *Advances in Neural Information Processing Systems 15*. MIT Press, 2003.
28. L. Wang, C. Yang, and J. Feng. On learning with dissimilarity functions. In *Proceedings of the 24th international conference on Machine learning*, pages 991 – 998, 2007.

A Weakly Good Similarity Functions

We show here that for any $\gamma > 0$, Definition 5 is enough to imply weak learning. In particular, first, determine if the distribution is noticeably skewed towards positive or negative examples: if so, weak-learning is immediate (output all-positive or all-negative respectively). Otherwise, draw a sufficiently large set S^+ of positive examples and set S^- of negative examples. Then, for each x, consider $\tilde{\gamma}(x) = \frac{1}{2}\left[\mathbf{E}_{x' \in S^+}[K(x,x')] - \mathbf{E}_{x' \in S^-}[K(x,x')]\right]$. 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.) We can then prove the following result:

Theorem 2 If K is a weakly γ-good similarity function, then with probability at least $1-\delta$, the above algorithm using sets S^+, S^- of size $\frac{64}{\gamma^2}\ln\left(\frac{64}{\gamma\delta}\right)$ yields a classifier with error at most $\frac{1}{2} - \frac{3\gamma}{128}$.

Proof. First, we assume the algorithm initially draws a sufficiently large sample such that if the distribution is skewed with probability mass greater than $\frac{1}{2}+\alpha$ on positives or negatives for $\alpha = \frac{\gamma}{32}$, then with probability at least $1-\delta/2$ the algorithm notices the bias and weak-learns immediately (and if the distribution is less skewed than $\frac{1}{2} \pm \frac{3\gamma}{128}$, with probability $1-\delta/2$ it does not incorrectly halt in this step). In the following, then, we may assume the distribution P is less than $(\frac{1}{2}+\alpha)$-skewed, and let us define P' to be P reweighted to have probability mass exactly $1/2$ on positive and negative examples. Thus, Definition 5 is satisfied for P' with margin at least $\gamma - 4\alpha$.

For each x define $\gamma(x)$ as $\frac{1}{2}\mathbf{E}_{x'}[K(x,x')|y(x')=1] - \frac{1}{2}\mathbf{E}_{x'}[K(x,x')|y(x')=-1]$ and notice that Definition 5 implies that $\mathbf{E}_{x\sim P'}\left[y(x)\gamma(x)\right] \geq \gamma/2 - 2\alpha$. Consider now the probabilistic prediction function g defined as $g(x)=1$ with probability $\frac{1+\gamma(x)}{2}$ and $g(x)=-1$ with probability $\frac{1-\gamma(x)}{2}$. We clearly have that for a fixed x,

$$\Pr_g(g(x) \neq y(x)) = \frac{y(x)(y(x)-\gamma(x))}{2},$$

which then implies that $\Pr_{x\sim P',g}(g(x) \neq y(x)) \leq \frac{1}{2} - \frac{1}{4}\gamma - \alpha$. Now notice that in our algorithm we do not use $\gamma(x)$ but an estimate of it $\tilde{\gamma}(x)$, and so the last step of the proof is to argue that this is good enough. To see this, notice first that d is large enough so that for any fixed x we have $\Pr_{S^+,S^-}\left(|\gamma(x) - \tilde{\gamma}(x)| \geq \frac{\gamma}{4} - 2\alpha\right) \leq \frac{\gamma\delta}{32}$. This implies $\Pr_{x\sim P'}\left(\Pr_{S^+,S^-}\left(|\gamma(x) - \tilde{\gamma}(x)| \geq \frac{\gamma}{4} - 2\alpha\right)\right) \leq \frac{\gamma\delta}{32}$, so

$$\Pr_{S^+,S^-}\left(\Pr_{x\sim P}\left(|\gamma(x) - \tilde{\gamma}(x)| \geq \frac{\gamma}{4} - 2\alpha\right) \geq \frac{\gamma}{16}\right) \leq \delta/2.$$

This further implies that with probability at least $1-\delta/2$ we have $\mathbf{E}_{x\sim P'}\left[y(x)\tilde{\gamma}(x)\right] \geq \left(1 - \frac{\gamma}{16}\right)\frac{\gamma}{4} - 2\frac{\gamma}{16} \geq \frac{7\gamma}{64}$. Finally using a reasoning similar to the one above (concerning the probabilistic prediction function based on $\gamma(x)$), we obtain that with probability at least $1-\delta/2$ the error of the probabilistic classifier based on $\tilde{\gamma}(x)$ is at most $\frac{1}{2} - \frac{7\gamma}{128}$ on P', which implies the error over P is at most $\frac{1}{2} - \frac{7\gamma}{128} + \alpha = \frac{1}{2} - \frac{3\gamma}{128}$. □