

Theory and Algorithm for Learning with Dissimilarity Functions

Liwei Wang (wanglw@cis.pku.edu.cn)

Key Laboratory of Machine Perception, MOE
School of Electronics Engineering and Computer Science, Peking University
Beijing, 100871, P.R.China

Masashi Sugiyama (sugi@cs.titech.ac.jp)

Department of Computer Science, Tokyo Institute of Technology
2-12-1, O-okayama, Meguro-ku, Tokyo, 152-8552, Japan

Cheng Yang (yangch@cis.pku.edu.cn)

Key Laboratory of Machine Perception, MOE
School of Electronics Engineering and Computer Science, Peking University
Beijing, 100871, P.R.China

Kohei Hatano (hatano@i.kyushu-u.ac.jp)

Department of Informatics, Kyushu University
744 Motooka, Nishi-ku, Fukuoka-city, 819-0395, Japan

Jufu Feng (fjf@cis.pku.edu.cn)

Key Laboratory of Machine Perception, MOE
School of Electronics Engineering and Computer Science, Peking University
Beijing, 100871, P.R.China

Abstract

We study the problem of classification when only a dissimilarity function between objects is accessible. That is, data samples are represented not by feature vectors but in terms of their pairwise dissimilarities. We establish sufficient conditions for dissimilarity functions to allow building accurate classifiers. The theory immediately suggests a learning paradigm: construct an ensemble of simple classifiers each depending on a pair of examples, then find a convex combination of them to achieve a large margin. We next develop a practical algorithm referred to as Dissimilarity based Boosting (DBoost) for learning with dissimilarity functions under the theoretical guidance. Experiments on a variety of databases demonstrate that the DBoost algorithm is promising for several dissimilarity measures widely used in practice.

Keywords

Dissimilarity functions, Boosting, Sample complexity

1 Introduction

In classification problems, objects are often represented by feature vectors in a Euclidean space. The Euclidean feature space provides much more analytical tools for classification than other representations. However, such a representation requires the selection of features, which is usually difficult and domain dependent. For example, in the area of fingerprint analysis, it took scientists for more than one hundred years to discover useful features for fingerprint recognition (Maltoni et al., 2003). It is not clear even today what kind of features have good discrimination ability for human face recognition, and existing feature extraction algorithms are not reliable or accurate enough (Zhao et al., 2003).

An alternative way is to describe the patterns using dissimilarity functions. Dissimilarity is a function that reflects the “distance” between two objects but with no restrictions on its mathematical properties. For some applications such as image retrieval, dissimilarity representation has the advantage that it is more convenient to define such a measure than a set of meaningful features (Jacobs et al., 2000; Jain and Zongker, 1997). A number of image dissimilarities have been proposed in the literature (Simard et al., 1993; Huttenlocher et al., 1993; Li et al., 2002; Wang et al., 2005) and successfully used in real-world applications. Dissimilarity functions can also be defined on strings, graphs and other structured objects (Gärtner, 2003; Saigo et al., 2004). This procedure thus provides a bridge between classical and structural approaches to pattern classification (Graepel et al., 1999; Goldfarb, 1985).

The simplest method to classify objects in dissimilarity representations is the nearest neighbor (NN) rule. NN has an appealing asymptotic property that its error rate converges to the Bayes optimal risk (Hart and Cover, 1967). However, the rate of convergence of NN could be very slow (Fukunaga, 1990), and it is observed in practice that NN is sensitive to the choice of dissimilarity measures and noise (Breiman et al., 1984).

In contrast to NN, there are several algorithms that take into account the global information as well. One type of methods first embeds the data into a (possibly pseudo) Euclidean space, then applies traditional Euclidean classification algorithms, with modifications adapted to the pseudo Euclidean if necessary (Graepel et al., 1999). Another type of methods explicitly constructs feature representations of the objects via their (dis)similarities to a set of prototypes, and then runs standard linear separator algorithms like Support Vector Machines (Vapnik, 1998) in the new space (Balcan et al., 2004, 2006; Pekalska and Duin, 2002). All these algorithms demonstrate superior performance to NN on a number of datasets.

In accordance to the progress made in algorithm development, a theoretical foundation of learning with dissimilarities is needed. In some cases, a theory of learning can be given in the form of sufficient conditions for efficient learning. The kernel theory is such an example—the theory states that large margin is a sufficient condition for good kernels: If the data are well separated with a large margin in an implicit high dimensional space induced by the kernel, then the kernel is good. That is, there exists a learning algorithm which can generate a classifier having a low generalization error with a small number of training examples. Also important in practice is that the large margin condition implies

learning algorithms that are computationally efficient. All the algorithms that only involve the inner product in the original input space can be kernelized by replacing the inner product by a positive semi-definite kernel.

Recently, Balcan and Blum (2006) developed a theory of learning with *similarities* in this way. They defined a notion of what it means for a pairwise function to be a good similarity function for a learning problem. They showed that their definition is sufficient to allow one to learn well and captures the standard notion of a good kernel with some degradation on learning parameters (see also Srebro, 2007; Balcan et al., to appear, for details). This theory immediately suggests algorithms that use feature representation based on prototypes as described earlier, and therefore provides a theoretical explanation of their good empirical performances.

In this paper we develop a theory for learning with dissimilarity functions, in parallel to Balcan and Blum's results. We propose new sufficient conditions for a dissimilarity function that yield good learning guarantees. These sufficient conditions also suggest a computationally efficient algorithm which is a boosting type algorithm that combines an ensemble of simple classifiers of special forms. We then make the algorithm more suitable for practical usage. An advantage of our theory and algorithm is that they are applicable to unbounded dissimilarity functions, while previous results deal with normalized similarity measures.

The paper is organized as follows: We describe our theory in Section 2. In Section 3, a practical algorithm called DBoost is proposed for learning with dissimilarity functions as a consequence of the theory. We provide experimental evidence of the benefits of our algorithm in Sections 4 and 5, and finally conclude in Section 6.

2 Theory

In this section we describe our theory of learning with dissimilarity functions. We propose sufficient conditions that have good learning guarantees and imply computationally efficient algorithms. We begin with a simple yet intuitively reasonable sufficient condition and then generalize it to incorporate more sophisticated cases.

2.1 Notations

By dissimilarity we mean any nonnegative bivariate function $d(x, x')$, where $x, x' \in X$, and X is an instance space. The axioms of a metric, i.e. reflectivity, symmetry and triangle inequality are not necessary for a dissimilarity function.

Labeled examples are represented by z, z', z'', \dots , where $z = (x, y)$, $x \in X$ and $y \in \{-1, 1\}$. The examples are drawn randomly and, either independently or conditionally independently, from the underlying distribution P of the problem over $X \times \{-1, 1\}$. These are always clear from the context. I denotes the indicator function, and $\text{sgn}(x) = 1$ if $x > 0$ and -1 otherwise.

2.2 Sufficient Conditions for Learning with Dissimilarity Functions

We propose in this subsection sufficient conditions for a dissimilarity function that are useful for learning.

2.2.1 Strong (ϵ, γ) -goodness

We first give a notion of good dissimilarity functions, which is quite intuitive. This definition expresses that if most examples are more likely to be *close* to random examples z' of the same class than to z'' of the opposite class, the dissimilarity function is good. More precisely, we use an *accuracy* parameter ϵ and a *margin* parameter γ to characterize the goodness of a dissimilarity function.

Definition 1 A dissimilarity function $d(x, x')$ is said to be strongly (ϵ, γ) -good for the learning problem, if at least $1 - \epsilon$ probability mass of examples z satisfy¹:

$$P(d(x, x') < d(x, x'') \mid y' = y, y'' = -y) \geq 1/2 + \gamma/2, \quad (1)$$

where the probability is over random examples $z' = (x', y')$ and $z'' = (x'', y'')$.

The notion of strongly (ϵ, γ) -good dissimilarity functions suggests a simple learning algorithm: draw pairs of examples of different labels, and vote according to which class the test example is more likely to be close to. This is summarized in the following theorem.

Theorem 2 If d is a strongly (ϵ, γ) -good dissimilarity function, then with probability at least $1 - \delta$ over the choice of $n = (4/\gamma^2) \ln(1/\delta)$ pairs of examples (z'_i, z''_i) with labels $y'_i = 1$, $y''_i = -1$, $i = 1, 2, \dots, n$, the following classifier

$$H(x) = \text{sgn} [f(x)],$$

where

$$f(x) = \frac{1}{n} \sum_{i=1}^n \text{sgn} [d(x, x''_i) - d(x, x'_i)],$$

has an error rate of no more than $\epsilon + \delta$. That is

$$P(yf(x) \leq 0) \leq \epsilon + \delta.$$

Proof The proof uses a technique in Balcan and Blum (2006). Let M be the set of examples satisfying (1). For any fixed $z = (x, y) \in M$,

$$\begin{aligned} & P(d(x, x') < d(x, x'') \mid y' = y, y'' = -y) \\ &= E(I(d(x, x') < d(x, x'')) \mid y' = y, y'' = -y) \\ &= \frac{1}{2} E(\text{sgn}[d(x, x'') - d(x, x')] \mid y' = y, y'' = -y) + \frac{1}{2}, \end{aligned}$$

¹In (1) using $\gamma/2$ is to make $0 \leq \gamma \leq 1$.

where the probability is over random examples $z' = (x', y')$ and $z'' = (x'', y'')$. Thus inequality (1) is equivalent to

$$E(\operatorname{sgn}[d(x, x'') - d(x, x')] \mid y' = y, y'' = -y) \geq \gamma.$$

The Chernoff bound then implies that:

$$P_{S_n}(y \cdot f(x) \leq 0) = P_{S_n}\left(y \cdot \frac{1}{n} \sum_{i=1}^n \operatorname{sgn}[d(x, x'_i) - d(x, x''_i)] \leq 0\right) \leq e^{-n\gamma^2/2},$$

where P_{S_n} denotes the probability over the choice of n pairs of training examples. Since the above inequality holds for every $z \in M$, we can take expectation over all $z \in M$, which results in that the expected error is at most $e^{-n\gamma^2/2}$, i.e.

$$E_{z \in M}[P_{S_n}(yf(x) \leq 0)] \leq e^{-n\gamma^2/2}.$$

Note that

$$E_{z \in M}[P_{S_n}(yf(x) \leq 0)] = P_{z \in M, S_n}(yf(x) \leq 0) = E_{S_n}[P_{z \in M}(yf(x) \leq 0)],$$

thus we have

$$E_{S_n}[P_{z \in M}(yf(x) \leq 0)] \leq e^{-n\gamma^2/2}.$$

Next using the Markov inequality we obtain that the probability that the error rate over the set M is larger than θ is at most $e^{-n\gamma^2/2}/\theta$ for arbitrary $\theta > 0$:

$$P_{S_n}[P_{z \in M}(yf(x) \leq 0) \geq \theta] \leq e^{-n\gamma^2/2}/\theta.$$

Finally, setting $\delta = e^{-n\gamma^2/2}/\theta$ and adding the ϵ probability of examples z not in M completes the proof. \blacksquare

2.2.2 (ϵ, γ, B) -goodness

Although Definition 1 and its suggested algorithm is natural, the notion of strong (ϵ, γ) -goodness may be too restrictive. Many good dissimilarity functions that can be used, in a more intellectually way, to learn well do not satisfy Definition 1. To illustrate this, we show in Fig.1 a simple one-dimensional example. In this learning problem, positive and negative examples are uniformly distributed in $[1/8, 9/8]$ and $[-9/8, -1/8]$ respectively. The dissimilarity function of two points x and x' used here is $d(x, x') = |x - x'|$. This problem should be perfectly learned with this dissimilarity function. However, for positive example $x \approx 1/8$ or negative example $x \approx -1/8$, the probability $P(d(x, x') < d(x, x'') \mid y' = y, y'' = -y)$ is not one. In fact, it is not difficult to show that for any positive example $x \in [1/8, 1/4]$ or negative example $x \in [-1/4, -1/8]$, we have

$$P(d(x, x') < d(x, x'') \mid y' = y, y'' = -y) = \frac{1}{2} + \frac{3/4}{2} \leq \frac{7}{8}.$$

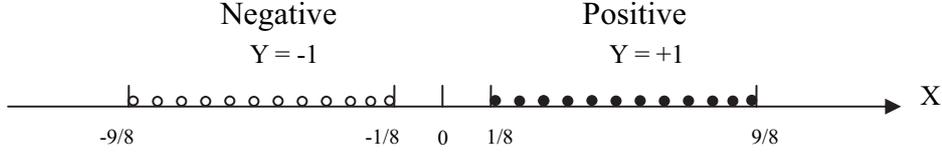


Figure 1: Positive examples are uniformly distributed in $[1/8, 9/8]$. Negative examples are uniformly distributed in $[-9/8, -1/8]$. $d(x, x') = |x - x'|$.

That is, $1/8$ probability mass of examples do not have a margin larger than $3/4$. Thus the dissimilarity function is not strongly $(1/8, \gamma)$ -good for any $\gamma > 3/4$.

Notice however, for any example (x, y) in the above problem, when randomly choosing x' (or x'') of the same (or opposite) class of x , if we only use the examples near the boundary, then we would have that $d(x, x') < d(x, x'')$ ($y' = y, y'' = -y$) holds for all examples (x, y) . To be concrete, if we draw x' according to a new distribution $\tilde{p}(x | y = 1)$ which is the uniform distribution on $[1/8, 3/8]$, and draw x'' according to $\tilde{p}(x | y = -1)$ which is the uniform distribution on $[-3/8, -1/8]$, we can learn a zero-error classifier. Therefore, with respect to the new distributions $\tilde{p}(x|y)$, the dissimilarity function is perfect: it is strongly $(0, 1)$ -good for the problem.

Generally, if we know the dissimilarity function is strongly good with respect to distributions $\tilde{p}(x|y = 1)$ and $\tilde{p}(x|y = -1)$, we could reweight the data as if they were generated from \tilde{p} and learn the classifier in the same way as described in Theorem 2.

In the following definition, a further step is made. We merely assume the existence of the new distributions, which are not necessarily known a priori. This definition therefore captures a broad class of dissimilarity functions. Later it will become clear that this assumption alone is sufficient to learn an accurate classifier.

Definition 3 Denote by $p(x | y = 1)$ and $p(x | y = -1)$ the conditional pdfs of the learning problem. A dissimilarity function d is said to be (ϵ, γ, B) -good for the learning problem if:

1. There exist two conditional pdfs $\tilde{p}(x | y = 1)$ and $\tilde{p}(x | y = -1)$ such that for all $x \in X$

$$\frac{\tilde{p}(x | y = 1)}{p(x | y = 1)} \leq \sqrt{B}, \quad \frac{\tilde{p}(x | y = -1)}{p(x | y = -1)} \leq \sqrt{B}.$$

2. At least $1 - \epsilon$ probability mass of examples z satisfy

$$\tilde{P}(d(x, x') < d(x, x'') | y' = y, y'' = -y) \geq 1/2 + \gamma/2, \quad (2)$$

where \tilde{P} is the probability with respect to $\tilde{p}(x'|y')$ and $\tilde{p}(x''|y'')$. That is,

$$\begin{aligned} & \tilde{P}(d(x, x') < d(x, x'') | y' = y, y'' = -y) \\ &= \iint I[d(x, x') < d(x, x'')] \tilde{p}(x' | y'=y) \tilde{p}(x'' | y''=-y) dx' dx''. \end{aligned}$$

The next theorem says that (ϵ, γ, B) -goodness guarantees the existence of a low-error large-margin classifier, which is a convex combination of the base classifiers.

Theorem 4 *If d is a (ϵ, γ, B) -good dissimilarity function, then with probability at least $1 - \delta$ over the choice of $n = 16B^2/\gamma^2 \ln(1/\delta)$ pairs of examples $((x'_i, 1), (x''_i, -1))$, $i = 1, 2, \dots, n$, there exists a convex combination classifier $f(x)$ of n base classifiers $h_i(x)$:*

$$f(x) = \sum_{i=1}^n \alpha_i h_i(x), \quad \sum \alpha_i = 1, \quad \alpha_i \geq 0,$$

where

$$h_i(x) = \text{sgn} [d(x, x''_i) - d(x, x'_i)],$$

such that the error rate of the combined classifier at margin $\gamma/2B$ is at most $\epsilon + \delta$. That is,

$$P(y \cdot f(x) \leq \gamma/2B) \leq \epsilon + \delta.$$

Proof Denote²

$$w_1(x) = \frac{\tilde{p}(x | y = 1)}{p(x | y = 1)}, \quad w_{-1}(x) = \frac{\tilde{p}(x | y = -1)}{p(x | y = -1)}.$$

Let M be the set of examples satisfying (2). For a fixed $z = (x, y) \in M$,

$$\begin{aligned} & \tilde{P}(d(x, x') < d(x, x'')) | y' = y, y'' = -y \\ &= \iint \tilde{p}(x' | y' = y) \tilde{p}(x'' | y'' = -y) I[d(x, x') < d(x, x'')] dx' dx'' \\ &= \iint w_y(x') w_{-y}(x'') p(x' | y' = y) p(x'' | y'' = -y) \left\{ \frac{\text{sgn}[d(x, x'') - d(x, x')] + 1}{2} \right\} dx dx' \\ &= \frac{1}{2} E \{ w_y(x') w_{-y}(x'') \text{sgn}[d(x, x'') - d(x, x')] | y' = y, y'' = -y \} + \frac{1}{2}. \end{aligned}$$

Hence

$$\tilde{P}(d(x, x') < d(x, x'') | y' = y, y'' = -y) \geq 1/2 + \gamma/2$$

is equivalent to

$$E \{ w_y(x') w_{-y}(x'') \text{sgn}[d(x, x'') - d(x, x')] | y' = y, y'' = -y \} \geq \gamma.$$

Note that $0 \leq w_1(x') w_{-1}(x'') \leq B$, the above inequality together with the Hoeffding inequality implies that:

$$P \left\{ y \frac{1}{n} \sum_{i=1}^n w_1(x'_i) w_{-1}(x''_i) \text{sgn}[d(x, x''_i) - d(x, x'_i)] \leq \gamma/2 \right\} \leq e^{-n\gamma^2/8B^2}$$

² $w_1(x)$ and $w_{-1}(x)$ are called the *importance* in the context of covariate shift adaptation (Shimodaira, 2000; Sugiyama et al., 2007).

Let $\alpha_i = \frac{w_1(x'_i) w_{-1}(x''_i)}{\sum_j w_1(x'_j) w_{-1}(x''_j)}$, we have:

$$P \left\{ y \sum_{i=1}^n \alpha_i \operatorname{sgn} [d(x, x''_i) - d(x, x'_i)] \leq \frac{n\gamma/2}{\sum_{j=1}^n w_1(x'_j) w_{-1}(x''_j)} \right\} \leq e^{-n\gamma^2/8B^2}.$$

Since $w_1(x'_i)w_{-1}(x''_i) \leq B$, we obtain

$$P \left\{ y \cdot \sum_{i=1}^n \alpha_i \operatorname{sgn} [d(x, x''_i) - d(x, x'_i)] \leq \gamma/2B \right\} \leq e^{-n\gamma^2/8B^2}.$$

Taking expectation over all $z \in M$ then using the Markov inequality as in the proof of Theorem 2, we complete the proof. \blacksquare

This theorem suggests a simple algorithm for learning with a (ϵ, γ, B) -good dissimilarity function d : First draw a set S_1 that contains $n = O(\frac{B^2}{\gamma^2} \ln 1/\delta)$ pairs of examples $((x'_i, 1), (x''_i, -1))$, $i = 1, 2, \dots, n$, and then construct n base classifiers

$$h_i(x) = \operatorname{sgn} [d(x, x''_i) - d(x, x'_i)].$$

It is guaranteed that with probability $1 - \delta$ there exists a low-error and large-margin classifier, which is a convex combination of these $h_i(x)$. Boosting would be natural for learning this large-margin voting classifier—thus one draws an additional set of examples S_2 , uses Boosting to learn the combination coefficients α_i , and obtains the final classifier

$$H(x) = \operatorname{sgn} \left[\sum_{i=1}^n \alpha_i h_i(x) \right].$$

In order that the final classifier $H(x)$ has an error rate at most $\epsilon + \epsilon_1$ with probability at least $1 - \delta$, the size of the second training set S_2 can be set as³ $\tilde{O}(\frac{B^2}{\epsilon_1^2 \gamma^2})$ according to the margin bound for convex combination classifiers (Schapire et al., 1998; Wang et al., 2008). The total amount of examples needed to achieve such a learning guarantee is $\tilde{O}(\frac{B^2}{\gamma^2 \epsilon_1^2})$ if we set $\delta = \epsilon_1$.

2.2.3 Generalized (ϵ, γ, B) -Goodness

In this subsection we present a generalization of the (ϵ, γ, B) -goodness. Recall in Definition 3 that for a (ϵ, γ, B) -good dissimilarity function d , most examples $z = (x, y)$ satisfy

$$\tilde{P}(d(x, x') - d(x, x'') < 0 \mid z, y' = y, y'' = -y) \geq 1/2 + \gamma/2,$$

³In \tilde{O} we hide the negligible logarithm terms of the learning parameters

where \tilde{P} is the probability with respect to two (unknown) pdfs $\tilde{p}(x'|y')$ and $\tilde{p}(x''|y'')$. A more broader class of dissimilarity functions would be that most examples $z = (x, y)$ satisfy

$$\tilde{P}(d(x, x') - d(x, x'') < v \mid z, y' = y, y'' = -y) \geq 1/2 + \gamma/2$$

for some threshold v , which may depend on the example pair (x', x'') .

Definition 5 Denote by $p(x \mid y = 1)$ and $p(x \mid y = -1)$ the conditional pdfs of the learning problem. A dissimilarity function d is said to be generalized (ϵ, γ, B) -good for the learning problem if:

1. There exist two conditional pdfs $\tilde{p}(x \mid y = 1)$ and $\tilde{p}(x \mid y = -1)$ such that for all $x \in X$

$$\frac{\tilde{p}(x \mid y = 1)}{p(x \mid y = 1)} \leq \sqrt{B}, \quad \frac{\tilde{p}(x \mid y = -1)}{p(x \mid y = -1)} \leq \sqrt{B}.$$

2. There exists a threshold $v(x', x'')$ such that at least $1 - \epsilon$ probability mass of examples z satisfy

$$\tilde{P}(d(x, x') - d(x, x'') < v(x', x'') \mid y' = y, y'' = -y) \geq 1/2 + \gamma/2, \quad (3)$$

where \tilde{P} is the probability with respect to $\tilde{p}(x'|y')$ and $\tilde{p}(x''|y'')$.

The learning guarantee of the generalized (ϵ, γ, B) -good dissimilarity functions is the same as that of the (ϵ, γ, B) -good dissimilarities if the threshold is known.

Theorem 6 Let d be a generalized (ϵ, γ, B) -good dissimilarity function. Assume that the threshold $v(x', x'')$ is known. Then with probability at least $1 - \delta$ over the choice of $n = 16B^2/\gamma^2 \ln(1/\delta)$ pairs of examples (z'_i, z''_i) with labels $y'_i = 1, y''_i = -1, i = 1, 2, \dots, n$, there exists a convex combination classifier $f(x)$ of n base classifiers $h_i(x)$:

$$f(x) = \sum_{i=1}^n \alpha_i h_i(x), \quad \sum \alpha_i = 1, \quad \alpha_i \geq 0,$$

where

$$h_i(x) = \text{sgn}[d(x, x''_i) - d(x, x'_i) + v(x', x'')]$$

such that the error rate of the combined classifier at margin $\gamma/2B$ is at most $\epsilon + \delta$.

Proof The proof follows the proof of Theorem 4 by replacing $\text{sgn}[d(x, x'') - d(x, x')]$ with $\text{sgn}[d(x, x'') - d(x, x') + v(x', x'')]$. ■

A generalized (ϵ, γ, B) -good dissimilarity function guarantees efficient learning if the threshold is given. In practice, it is difficult to know the threshold a priori. However, one can learn the thresholds and the linear coefficients simultaneously in the Boosting framework. Namely, we draw a set S_1 of n pairs of examples $((x', 1), (x'', -1))$ $i = 1, 2, \dots, n$,

and a training set S_2 . Then we use Boosting to learn the thresholds $v(x', x'')$ and the coefficients α_i so that the combined classifier

$$f(x) = \sum_{i=1}^n \alpha_i h_i(x), \quad \sum \alpha_i = 1, \quad \alpha_i \geq 0,$$

$$h_i(x) = \text{sgn} [d(x, x'_i) - d(x, x''_i) + v(x', x'')]$$

has low-error and large-margin on the training set S_2 . We will describe this algorithm in detail and make it more practical in Section 3.

2.2.4 $(\epsilon, \gamma, B, \eta)$ -Goodness

We propose a further generalization of the previous goodness definitions by weakening the first condition in Definition 5. Here we do not require there are uniform upper bounds of $\frac{\tilde{p}(x|y)}{p(x|y)}$. We only need that $\frac{\tilde{p}(x|y)}{p(x|y)} \leq B$ for most examples. This new definition of goodness can be applied to a much broader class of dissimilarity functions and contains all previous goodness notions as special cases.

Definition 7 Denote by $p(x | y = 1)$ and $p(x | y = -1)$ the conditional pdfs of the learning problem. A dissimilarity function d is said to be $(\epsilon, \gamma, B, \eta)$ -good for the learning problem if:

1. There exist two conditional pdfs $\tilde{p}(x | y = 1)$ and $\tilde{p}(x | y = -1)$ such that

$$\frac{\tilde{p}(x | y = 1)}{p(x | y = 1)} \leq \sqrt{B}, \quad \frac{\tilde{p}(x | y = -1)}{p(x | y = -1)} \leq \sqrt{B}$$

hold for at least $1 - \eta$ probability mass of examples (x, y) .

2. There exists a threshold $v(x', x'')$ such that at least $1 - \epsilon$ probability mass of examples z satisfy

$$\tilde{P}(d(x, x') - d(x, x'') < v(x', x'') | y' = y, y'' = -y) \geq 1/2 + \gamma/2, \quad (4)$$

where \tilde{P} is the probability with respect to $\tilde{p}(x'|y')$ and $\tilde{p}(x''|y'')$.

The learning guarantee of a $(\epsilon, \gamma, B, \eta)$ -good dissimilarity function is the same as the (ϵ, γ, B) -good dissimilarity only up to a constant factor.

Theorem 8 If d is a $(\epsilon, \gamma, B, \eta)$ -good dissimilarity function, then with probability at least $1 - \delta$ over the choice of $n = 16B^2/\gamma^2 \ln(2/\delta)$ pairs of examples (z'_i, z''_i) with labels $y'_i = 1, y''_i = -1, i = 1, 2, \dots, n$, there exists a convex combination classifier $f(x)$ of n base classifiers $h_i(x)$:

$$f(x) = \sum_{i=1}^n \alpha_i h_i(x), \quad \sum \alpha_i = 1, \quad \alpha_i \geq 0,$$

where

$$h_i(x) = \text{sgn} [d(x, x_i'') - d(x, x_i') + v(x_i', x_i'')]$$

such that the error rate of the combined classifier at margin $\gamma/2B$ is at most $\epsilon + \delta$, provided $\eta \leq \frac{\gamma^2 \delta}{64B^2 \ln(2/\delta)}$ and the threshold is known.

Proof The proof is almost the same as Theorem 6. The only difference is that by our assumption, when choosing $n = 16B^2/\gamma^2 \ln(2/\delta)$ pairs of examples, the probability that there contains a “bad” example is at most $\delta/2$, where a bad example $z' = (x', y')$ or $z'' = (x'', y'')$ means that $\frac{\tilde{p}(x'|y'=1)}{p(x'|y'=1)} > \sqrt{B}$ or $\frac{\tilde{p}(x''|y''=-1)}{p(x''|y''=-1)} > \sqrt{B}$ respectively. ■

2.3 Discussions of the Sufficient Conditions

In this subsection we compare our results with existing theory on learning with (dis)similarity functions, and study possible extension of the proposed sufficient conditions.

2.3.1 Comparison to Previous Theory on Learning with Similarity Functions

We first point out that our results of dissimilarity functions can be easily extended to similarity functions. Let $s(x, x')$ denote a similarity function. Just replacing $d(x, x') < d(x, x'')$ by $s(x, x') > s(x, x'')$ in all definitions and theorems gives the theory for similarity functions. Therefore, the theory is a unified framework for learning with similarity and dissimilarity functions.

In Balcan and Blum (2006) theory, a similarity function s is said to be (ϵ, γ) -good for a (deterministic label) learning problem if there exists a weighting function $w(x) \in [0, 1]$, such that at least $1 - \epsilon$ probability mass of examples x satisfy

$$E [w(x')s(x, x')|y' = y] \geq E [w(x')s(x, x')|y' = -y] + \gamma.$$

For a (ϵ, γ) -good similarity function, there is a simple learning approach. First draw a set of examples $\{x_1, x_2, \dots, x_n\}$. Then construct a n -dimensional feature vector of each object x as

$$(s(x, x_1), s(x, x_2), \dots, s(x, x_n)).$$

It can be shown that with high probability there is a large-margin low-error linear separator in the n -dimensional feature space. So linear SVM can be used to learn the classifier in the feature space with a new set of examples.

On the practical side, the theory of Balcan and Blum implies a SVM type algorithm, while ours suggests a Boosting type algorithm. On the theoretical side, their notion of good similarity functions and ours are different sufficient conditions for learning with (dis)similarity functions. That is, neither is a subset of the other, as described in the following proposition.

Proposition 9

1. For every γ and B , there is a similarity function $s(\cdot, \cdot)$ and a learning problem P , such that $s(\cdot, \cdot)$ is $(0, \gamma, B)$ -good for P in our sense, but not $(0, \gamma/B)$ -good in Balcan and Blum's sense.
2. For every γ , there is a similarity function $s(\cdot, \cdot)$ and a learning problem P , such that $s(\cdot, \cdot)$ is $(0, \gamma)$ -good for P in Balcan and Blum's sense, but not $(0, B\gamma, B)$ -good in our sense for any B .

Very recently, Balcan et al. (2008) proposed an improved sufficient condition for learning with similarity functions: s is said to be a (ϵ, γ, τ) -good similarity function for a learning problem P if there is a random indicator variable R (depending on x') such that $E[yy's(x, x')|R] \geq \gamma$ and $\Pr(R) \geq \tau$. Here R can be understood as a (stochastic) membership function on x' , indicating whether it is “important” or not.

This new notion of good similarity function and our definition of goodness are still different sufficient conditions, as shown in the following proposition.

Proposition 10

1. For every γ and B , there is a similarity function $s(\cdot, \cdot)$ and a learning problem P , such that $s(\cdot, \cdot)$ is $(0, \gamma, B)$ -good for P in our sense, but not $(0, \gamma', \tau')$ -good in Balcan et al. improved sense for any γ' and τ' such that $\gamma'\tau' \geq \gamma/B$.
2. For every γ , there is a similarity function $s(\cdot, \cdot)$ and a learning problem P , such that $s(\cdot, \cdot)$ is $(0, \gamma, \tau)$ -good for P in Balcan et al. improved sense, but not $(0, B\gamma', B)$ -good in our sense for any B and any $\gamma' \geq \max(\gamma, \tau)$.

Proof This is immediate from the previous proposition and the following relation between improved (ϵ, γ, τ) -goodness and the (ϵ, γ) -goodness described in (Balcan et al., 2008): A (ϵ, γ, τ) -good similarity function is also a $(\epsilon, \gamma\tau)$ -good similarity function; And a (ϵ, γ') -good similarity function is also a (ϵ, γ, τ) similarity function, where $\gamma' \geq \max(\gamma, \tau)$. ■

To conclude the comparison, our (ϵ, γ, B) -goodness implies that the “order” of the (dis)similarity is important. That is, data from the same class are closer than those from different classes. But how much closer is not crucial. On the other hand, the (ϵ, γ) -goodness and its improvement of Balcan et al. imply that the average value of the similarity is important. From a practical viewpoint, if the user has some confidence that the data from the same class are more likely to be closer to each other, our result and the suggested Boosting type algorithm apply. This is especially suitable for the applications in which an appropriate scaling of the (dis)similarities is difficult or expensive. In case the (dis)similarities are well scaled so that there is a significant difference of the within-class and between-class (average) similarity, the SVM type algorithm suggested by the (ϵ, γ) -goodness apply.

2.3.2 Pseudo Good Dissimilarity Functions

We require in our definitions of good dissimilarity functions that most of the examples (at least $1 - \epsilon$ probability mass) are more likely to be close to a random example of the same class than to an example of the opposite class. Another natural but weaker notion would be the following:

Definition 11 *A dissimilarity function $d(x, x')$ is said to be pseudo γ -good for the learning problem, if*

$$P(d(x, x') < d(x, x'') \mid y' = y, y'' = -y) \geq 1/2 + \gamma/2, \quad (5)$$

where the probability is taken over the random examples $z = (x, y)$, $z' = (x', y')$ and $z'' = (x'', y'')$.

Note that the pseudo γ -goodness is a weaker notion than strong (ϵ, γ) -goodness, as described in the following proposition.

Proposition 12 *If a dissimilarity function is strongly (ϵ, γ) -good for a learning problem, then it is also pseudo γ' -good, where $\gamma' = (1 - \epsilon)\gamma - \epsilon$ (if $\gamma' \geq 0$).*

Proof The proposition is immediate by noting that the left hand side of Eq.(5) is the expectation of the left hand side of Eq.(1) over $z = (x, y)$. ■

One might expect that the pseudo goodness would also imply learnability, possibly in a weak sense. However, the next proposition shows that the majority voting scheme given in previous theorems, in general, does not guarantee any learnability, even in the weakest sense. This result means that our sufficient conditions for efficient learning with dissimilarity functions may not be weakened too much.

Proposition 13 *For any pseudo γ -good ($0 < \gamma < 1/2$) dissimilarity function d , there exists a learning problem such that even we have infinitely many pairs of training examples (z'_i, z''_i) with labels $y'_i = 1, y''_i = -1, i = 1, 2, \dots, n$, the error rate of the voting classifier is higher than $1/2$. That is,*

$$P(y \cdot f(x) < 0) > 1/2,$$

where

$$f(x) = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n \text{sgn}[d(x, x''_i) - d(x, x'_i)].$$

Proof For a fixed example $z = (x, y)$, when $n \rightarrow \infty$, the law of large numbers gives that $f(x)$ converges in probability to

$$E \{ \text{sgn}[d(x, x'') - d(x, x')] \mid z, y' = 1, y'' = -1 \}.$$

Note further that

$$\begin{aligned} E \{ \operatorname{sgn} [d(x, x'') - d(x, x')] \mid z, y' = 1, y'' = -1 \} \\ = 2P \{ d(x, x') < d(x, x'') \mid z, y' = 1, y'' = -1 \} - 1. \end{aligned}$$

An error occurs, i.e. $yf(x) < 0$ if

$$P \{ d(x, x') < d(x, x'') \mid z, y' = y, y'' = -y \} < 1/2.$$

Denote

$$g(z) = P (d(x, x') < d(x, x'') \mid z, y' = y, y'' = -y).$$

For $0 < \gamma < 1/2$, it is easy to construct a distribution⁴ such that the following two inequalities hold simultaneously:

$$E [g(z)] \geq 1/2 + \gamma/2,$$

$$P (g(z) < 1/2) > 1/2.$$

The first inequality is equivalent to (3), meaning that the dissimilarity function is pseudo γ -good for the problem. The second inequality implies that the error rate of the voting classifier is larger than $1/2$. ■

3 The DBoost Algorithm

In this section, we slightly modify the algorithm suggested by our theory to make it more suitable for practical usage. The proposed algorithm is essentially a Dissimilarity based Boosting, which will be referred to as DBoost.

Recall that under the theoretical guidance, to learn with a dissimilarity function d , one needs to draw two sets of examples. One set contains n pairs of examples $((x'_i, 1), (x''_i, -1))$, with which we construct the base classifiers

$$h_i(x) = \operatorname{sgn} [d(x, x''_i) - d(x, x'_i) + v_i], \quad i = 1, 2, \dots, n.$$

The other set of examples are used as training data for Boosting to learn the thresholds v_i and the combination coefficients α_i so that the voting classifier

$$f(x) = \sum_{i=1}^n \alpha_i h_i(x), \quad \sum \alpha_i = 1, \quad \alpha_i \geq 0$$

has a low-error and large-margin on the training set.

In practice however, the users often have only one fixed set of examples. So in the DBoost algorithm we try to make use of the data efficiently. Denote by S the set of

⁴For example: $P(g(z) = 1/2 - \delta) = 1/2 + \delta$, $P(g(z) = 1) = 1/2 - \delta$, for some small δ .

data the user has. The DBoost algorithm first constructs the pairs $(x'_i, 1), (x''_i, -1)$ by considering all possible pairs of examples with different labels in S . Then S is also served as the training set for Boosting to learn the final large-margin convex-combination classifier.

In the DBoost algorithm, we choose AdaBoost as the booster due to its good ability to generate large-margin classifiers (Freund and Schapire, 1996). The thresholds v_i and the coefficients α_i are learned by AdaBoost in a series of rounds. At the i th round, we have a distribution D_i over the training set S . The algorithm then searches for the example pair (x'_i, x''_i) and the threshold v_i so that the base classifier

$$h_i(x) = \text{sgn} [d(x, x''_i) - d(x, x'_i) + v_i]$$

has the minimum training error on S with respect to the distribution D_i . The (unnormalized) coefficient α_i is determined by this training error, and the distribution D_i is also updated accordingly (see Fig.3 for details). After T rounds, the DBoost outputs the final classifier

$$H(x) = \text{sgn} [f(x)],$$

where

$$f(x) = \sum_{i=1}^T \alpha_i h_i(x).$$

One difficulty in the above procedure is that at each round searching for the best example pair (x'_i, x''_i) over all possible pairs is computationally expensive. We solve this problem in DBoost by searching for the best pair not over the whole set of possible pairs, but only over a small number M of example pairs randomly selected at each round. The selection of these example pairs is according to the current distribution D_i . It is well known that as the round i increases, Boosting makes the distribution D_i put larger weights on the examples harder to classify. Therefore, the example pair would be two data near the classification boundary since they are the most difficult examples.

The DBoost algorithm is described in detail in Fig.2 and Fig.3. Fig.2 shows the subroutine of searching for the pair (x', x'') and the threshold v to construct the base classifier at a certain round of boosting. Fig.3 describes the boosting framework in DBoost which is essentially the AdaBoost algorithm. Here we just show the original version of the AdaBoost given by Freund and Schapire (1996). An improved version—RealBoost (Schapire and Singer, 1999) can often achieve better performance. RealBoost folds the coefficient α_t into the base classifier h_t , and hence h_t outputs a real number. For details please refer to Schapire and Singer (1999).

Input: Dissimilarity function $d(\cdot, \cdot)$
 Training set $S = \{(x_1, y_1), (x_2, y_2), \dots, (x_l, y_l)\}$
 Distribution D on the training examples

Initialization: $err^* = \infty$

for $m = 1$ **to** M **do**

1. Randomly select two examples x_+^m, x_-^m according to D , and construct the pair.
2. Find v^m so that the classifier

$$h^{(m)}(x) = \text{sgn} [d(x, x_+^m) - d(x, x_-^m) + v^m]$$
 has the minimum error rate err^m , where $err^m = \sum_{i=1}^l D(i)I [y_i \neq h^{(m)}(x_i)]$, and I is the indicator function.
3. **if** $err^m < err^*$,
 $err^* = err^m, x_+^* = x_+^m, x_-^* = x_-^m, v^* = v^m$
 endif

end

Output: The base classifier

$$h(x) = \text{sgn} [d(x, x_+^*) - d(x, x_-^*) + v^*].$$

Figure 2: The subroutine: BaseLearn

Input: Dissimilarity function $d(\cdot, \cdot)$
 $S = \{(x_1, y_1), (x_2, y_2), \dots, (x_l, y_l)\}$
 where $x_i \in X, y_i \in \{-1, 1\}$.

Initialization: $D_1(i) = 1/l$.

for $t = 1$ **to** T **do**

1. Call BaseLearn(d, S, D_t); The returned classifier is denoted as h_t .
2. Let

$$\alpha_t = \frac{1}{2} \ln \frac{1 + \gamma_t}{1 - \gamma_t},$$
 where $\gamma_t = \sum_{i=1}^l D_t(i) y_i h_t(x_i)$.
3. Update:

$$D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t},$$
 where Z_t is a normalization factor chosen so that D_{t+1} will be a distribution.

end

Output: The final Classifier

$$H(x) = \text{sgn} \left(\sum_{t=1}^T \alpha_t h_t(x) \right).$$

Figure 3: The main DBoost algorithm

4 Experiments on Learning with Dissimilarity Functions

In this section we perform experiments on algorithms that learn with dissimilarity functions. We compare DBoost to the nearest neighbor rule and the algorithm based on Balcan and Blum’s theory mentioned in the Introduction. The algorithms will be denoted by NN and LSVM for short, respectively. The details of the implementation are described below.

1. **DBoost:** For DBoost, there are two parameters to be specified. One is M in Fig.2, that is, the number of example pairs forming the search space. As discussed earlier, introducing M is to reduce the computational cost. We found that the value of M has no significant effect on the performance unless M is too small. So in all the experiments we simply set $M = 100$. The other parameter is T in Fig.3, i.e. the number of base classifiers generated in boosting. We set $T = 1000$ in all the experiments.
2. **NN:** We use the one-nearest neighbor (1NN) classifier.
3. **LSVM:** This approach is based on Balcan and Blum’s theory of learning with similarity functions. Given a similarity function s , we choose a set of prototypes $\{p_1, p_2, \dots, p_r\}$. For each training data x , calculate the similarity of x and the prototypes, and then construct the vector $(s(x, p_1), s(x, p_2), \dots, s(x, p_r))$. This vector is treated as the feature representation of x . The final step is to run linear SVM on this r -dimensional space to obtain the classifier.

As mentioned earlier, to have a learning guarantee for this algorithm, the similarity should be a normalized function. That is $|s| \leq 1$. So we need to transform the dissimilarity d to a normalized s . We consider

$$s = e^{-d^2/\sigma}.$$

The value of the parameter σ is tuned by cross validation on the training set. In our implementation, we randomly select twenty percent of the training examples as prototypes, and we run libsvm (Chang and Lin, 2001) to obtain the linear SVM classifier.

In the first set of experiments we study image classification. For some cases it is more convenient to directly define dissimilarities between images than to construct meaningful features. Many dissimilarity measures of images have been proposed in the literature. We adopt in this experiment three measures: the Tangent distance (Simard et al., 1993), the Fuzzy Image Metric (Li et al., 2002) and the Euclidean distance. We perform the experiments on the USPS database which consists of images of handwritten digits. The dataset has been partitioned into a fixed training set and a test set, consisting of 7291 and 2007 examples respectively.

Fig. 4 shows the results of the three algorithms with the three dissimilarity measures respectively. By using the Tangent distance, NN achieves the best performance. In fact, the Tangent distance is developed specifically for the handwritten-digit classification problem. It incorporates strong domain knowledge and is invariant to local within-class transformations (Simard et al., 1993). Therefore the Tangent distance determines a good local topology for handwritten digit images. If two images have a very small Tangent distance, they are highly likely to be in the same class.

The other two dissimilarities—the Fuzzy Image Metric and the Euclidean distance are more general purpose measures. The information of the local distance alone is not enough for an accurate classification in this handwritten digit problem. So the performance of NN with these two dissimilarities are not excellent. On the other hand, DBoost works well in these cases. This result implies that the Fuzzy Image Metric and the Euclidean distance are still good dissimilarity functions. In other words, they contain enough information to build accurate classifiers.

We then consider the dissimilarity with discrete (qualitative) values. For instance, when people make subjective evaluation of the similarity between images, only qualitative (discrete) values can be given. For example, *similar*, *average* and *dissimilar* are possible values of a three-level qualitative dissimilarity. We evaluate the algorithms on such qualitative measures. To conduct the experiments, Euclidean distances are quantized to 3 to 15 levels respectively. The results on the USPS datasets are shown in Fig. 5. It shows that even with the 3-level measures, for which the information of local topology are mostly lost, DBoost still has a low error rate.

We also evaluate the performance of the algorithms against noisy data. We add to the USPS images Gaussian white noise with different variances, and give the algorithms as input the Euclidean distances. The results are depicted in Fig. 6, showing that DBoost is the most robust to noise.

In the second set of experiments, we evaluate the algorithms on a variety of domains. We adopt twenty two benchmark datasets from the UCI repository (Asuncion and Newman, 2007). The aim is to see whether the DBoost algorithm works well for dissimilarity functions that are widely used in practice. Here we consider two dissimilarity measures: l_1 and l_∞ . In the experiments each dataset is used in a five-fold cross validation fashion. The datasets are described in Table 1.

The results are listed in Table 2. For each dataset, the algorithm that has the best performance and those that are comparable to the best according to the t-test at the significant level 0.01 are marked in boldface. From these results one can see that, on the whole, DBoost has the best performance with these two dissimilarities even though they are not good measures for the NN classifier.

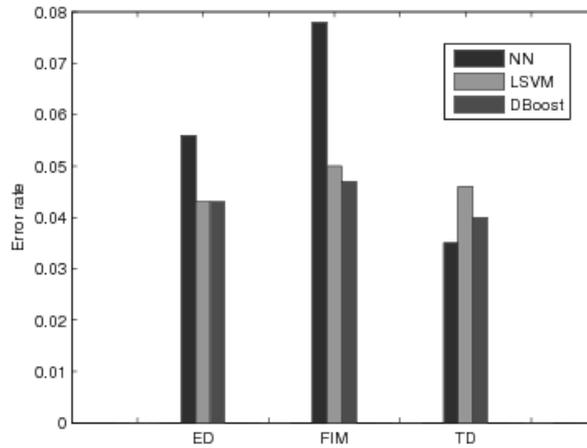


Figure 4: Comparison of the algorithms on USPS dataset using three dissimilarity functions: Euclidean distance (ED), Fuzzy Image Metric (FIM), and Tangent Distance (TD)

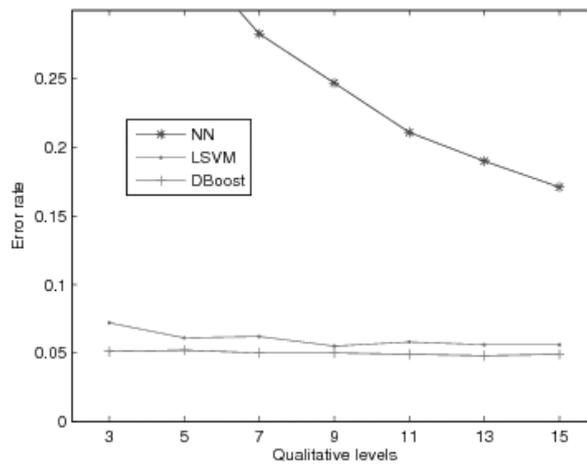


Figure 5: Comparison of the algorithms using quantized dissimilarity functions on the USPS database.

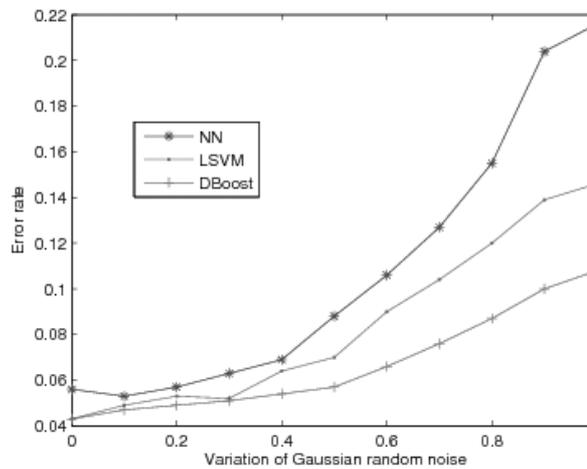


Figure 6: Comparison of the algorithms on the USPS database with noise.

Table 1: Description of the 22 datasets from UCI repository.

DATA SET	# CLASSES	# EXAMPLES	DATA SET	# CLASSES	# EXAMPLES
BALANCE	3	625	LETTER	26	20000
BREAST	2	699	LIVER	2	345
CLEVELAND	2	297	MONK1	2	556
DIABETES	2	768	MONK2	2	601
ECHO	2	106	MONK3	2	554
GERMAN	2	1000	SATIMAGE	6	6435
HAYS	3	160	VEHICLE	4	846
HEPATITIS	2	155	VOWEL	11	990
IMAGE	7	2310	WDBC	2	569
IONOSPHERE	2	351	WINE	3	178
IRIS	3	150	WPBC	2	194

Table 2: Comparison of the performances of the six algorithms on UCI datasets.

DATA SET	l_1			l_∞		
	NN	LSVM	DBoost	NN	LSVM	DBoost
BALANCE	20.2 ± 1.4	7.8 ± 2.9	5.6 ± 2.3	24.0 ± 3.9	6.1 ± 1.7	5.3 ± 2.6
BREAST	3.2 ± 1.7	2.5 ± 1.7	2.9 ± 1.8	5.0 ± 1.9	3.1 ± 2.1	3.4 ± 2.6
CLEVELAND	22.6 ± 5.4	17.8 ± 6.0	20.5 ± 5.5	26.2 ± 6.0	27.0 ± 4.3	24.9 ± 8.5
DIABETES	29.3 ± 1.7	23.8 ± 3.2	26.4 ± 2.8	29.0 ± 1.5	22.9 ± 2.8	27.2 ± 1.3
ECHO	21.7 ± 7.3	16.9 ± 3.8	12.3 ± 2.7	22.6 ± 7.8	16.9 ± 3.8	17.9 ± 3.9
GERMAN	30.4 ± 2.9	26.7 ± 2.2	27.3 ± 2.2	33.1 ± 2.9	30.1 ± 1.8	29.5 ± 3.4
HAYES	27.5 ± 7.4	16.2 ± 3.4	18.7 ± 3.2	32.5 ± 6.8	26.2 ± 4.7	21.3 ± 4.6
HEPATITIS	21.3 ± 9.3	18.4 ± 3.3	17.8 ± 9.6	19.1 ± 8.8	19.1 ± 6.6	19.1 ± 8.8
IMAGE	2.1 ± 0.6	3.4 ± 1.0	1.6 ± 0.4	3.3 ± 1.0	3.9 ± 1.0	2.7 ± 0.6
IONOSPHERE	10.8 ± 2.8	6.8 ± 2.7	5.4 ± 2.9	13.1 ± 1.6	7.7 ± 2.9	5.7 ± 3.9
IRIS	7.3 ± 4.3	4.0 ± 3.7	6.7 ± 3.4	4.7 ± 3.8	2.7 ± 2.8	4.0 ± 3.7
LETTER	4.7 ± 0.3	4.2 ± 0.2	5.0 ± 0.1	8.8 ± 0.5	4.1 ± 0.4	6.3 ± 0.3
LIVER	40.3 ± 6.7	31.6 ± 4.7	30.4 ± 6.4	41.4 ± 3.8	37.4 ± 4.0	31.2 ± 5.7
MONK1	20.3 ± 4.6	8.6 ± 3.7	0.0 ± 0.0	20.1 ± 4.9	19.0 ± 2.8	1.8 ± 1.4
MONK2	14.0 ± 3.4	24.1 ± 2.5	1.7 ± 1.8	14.0 ± 3.3	27.8 ± 3.9	4.5 ± 2.8
MONK3	19.1 ± 9.1	3.6 ± 0.9	2.3 ± 0.8	24.5 ± 1.4	6.5 ± 2.0	4.1 ± 1.0
SATIMAGE	9.4 ± 1.0	8.8 ± 0.9	8.0 ± 0.9	12.1 ± 1.6	9.6 ± 1.0	9.8 ± 0.8
VEHICLE	30.9 ± 2.4	28.3 ± 0.9	27.0 ± 2.4	32.6 ± 4.9	28.8 ± 1.6	22.6 ± 2.1
VOWEL	2.0 ± 1.2	8.5 ± 2.0	2.7 ± 1.5	2.6 ± 1.0	7.7 ± 2.4	3.6 ± 1.4
WDBC	4.2 ± 1.4	3.5 ± 0.6	2.3 ± 0.8	6.1 ± 1.4	5.4 ± 1.4	5.8 ± 2.5
WINE	4.5 ± 1.5	2.2 ± 2.3	3.9 ± 3.2	6.2 ± 3.7	1.7 ± 1.5	2.8 ± 2.8
WPBC	31.4 ± 3.8	23.7 ± 5.7	22.7 ± 4.1	37.1 ± 5.1	23.7 ± 5.7	26.3 ± 5.0

5 Experiments on Learning with Similarities

In this section we provide experimental results of algorithms on learning with a similarity function. As pointed out in Section 2.3, our theory and algorithm can be applied to similarities with only minor modifications. To be specific, in order to use the DBoost algorithm for a similarity function $s(x', x'')$, we only need to change the base classifier

$$h(x) = \text{sgn} [d(x, x_+^*) - d(x, x_-^*) + v^*]$$

in Fig. 2 to

$$h(x) = \text{sgn} [s(x, x_-^*) - s(x, x_+^*) - v^*].$$

We run experiments in the similarity setting with a music classification task. The goal is to evaluate the algorithms with a practical similarity measure. The data consists of 50 Japanese Pop songs and 50 Japanese traditional songs (used in a five-fold cross validation fashion). Each song is represented in the MIDI format, which contains a set of sequences of musical notes, cues, tones, volumes and so on. We extract the main melody from each song and convert it to a string, where each character corresponds to a sixteenth note or a sixteenth rest.

The similarity measure we use for a pair of songs is the length of the Longest Common Subsequence (LCS). The LCS of a string s and a string t is the longest (possibly nonconsecutive) sequence of characters which appear both in s and t . For example, given two strings $s = aabbcc$ and $t = abca$, the LCS of s and t is abc . The length of LCS is a natural measure to capture the similarity between two melodies since similar songs would share longer subsequences.

However, the similarity matrix based on the length of LCS is, in general, not positive semi-definite (Hagio, 2006). For example, consider the set of strings $\{a, b, ab, ba\}$. It is easy to check that the LCS-based similarity matrix is

$$\begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 2 \end{bmatrix},$$

which has a negative eigenvalue.

We evaluate four algorithms in this experiment. Besides NN, LSVM and DBoost, we also consider ordinary SVM for this non-positive similarity kernel. We adopt a common approach which adds to the similarity (kernel) matrix a positive diagonal to enforce it to be positive definite. Fig. 7 shows the performances of four algorithms. These results demonstrate that DBoost is promising for this simple and intuitive similarity measure.

6 Conclusions

In this work we gave sufficient conditions for dissimilarity functions to allow one to learn well. We showed that if most examples are more likely to be close to a randomly selected

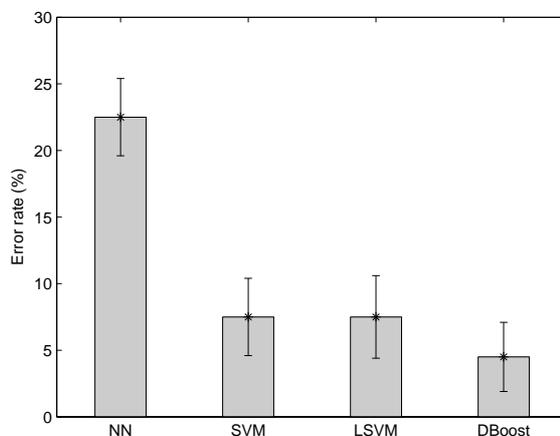


Figure 7: Results on music similarity data

example of the same class than to a random example of the other class, then there is a simple algorithm that can learn well with the dissimilarity measure. The random selection of the examples could be according to arbitrary probability distributions satisfying a mild condition. Therefore the sufficient condition captures a large class of dissimilarity functions. We also developed a more practical algorithm named DBoost under the theoretical guidance. DBoost learns a large-margin convex-combination classifier of a set of base classifiers, each of which depends only on the dissimilarities. Experimental results demonstrate that DBoost has good performance with several dissimilarity measures widely used in practice.

Acknowledgement

We thank Masayuki Takeda for kindly providing us Japanese songs data and we also thank Kazuhito Hagio for preprocessing them. This work was supported by NSFC(60775005, 60635030) and Global COE Program of Tokyo Institute of Technology.

References

- A. Asuncion and D. J. Newman. UCI machine learning repository, 2007.
- M.-F. Balcan and A. Blum. On a theory of learning with similarity functions. In William W. Cohen and Andrew Moore, editors, *International Conference on Machine Learning*, 2006.
- M.-F. Balcan, A. Blum, and N. Srebro. Improved guarantees for learning via similarity functions. In *21st Annual Conference on Learning Theory*, 2008.

- M.-F. Balcan, A. Blum, and N. Srebro. A theory of learning with similarity functions. *Machine Learning*, to appear.
- M.-F. Balcan, A. Blum, and S. Vempala. On kernels, margins, and low-dimensional mappings. In *International Workshop on Algorithmic Learning Theory*, 2004.
- M.-F. Balcan, A. Blum, and S. Vempala. Kernels as features: On kernels, margins, and low-dimensional mappings. *Machine Learning*, 65:79–94, 2006.
- L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth, 1984.
- C. C. Chang and C. J. Lin. *A library for support vector machines*. Available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>, 2001.
- Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *International Conference on Machine Learning*, 1996.
- K. Fukunaga. *Introduction to Statistical Pattern Recognition (Second Edition)*. Academic Press., 1990.
- T. Gärtner. A survey of kernels for structured data. *SIGKDD Explorations*, 5(1):S268–S275, 2003.
- L. Goldfarb. A new approach to pattern recognition. In L. N. Kanal and A. Rosenfeld, editors, *Progress in Pattern Recognition*, volume 2, pages 241–402. Elsevier, 1985.
- T. Graepel, R. Herbrich, P. Bollmann-Sdorra, and K. Obermayer. Classification on pairwise proximity data. *Advances in Neural Information Processing Systems*, 1999.
- K. Hagio. Design and evaluation of string similarity measure based kernels. Bachelor thesis, Department of Electrical Engineering and Computer Science, Kyushu University, March 2006. In Japanese.
- P. Hart and T. Cover. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, (13):21–27, 1967.
- D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge. Comparing images using the hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863, 1993.
- D. W. Jacobs, D. W., and Y. Gdalyahu. Classification with nonmetric distances: Image retrieval and class representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(6):583–560, 2000.
- A. K. Jain and D. E. Zongker. Representation and recognition of handwritten digits using deformable templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(12):1386–1391, 1997.

- J. Li, G. Chen, and Z. Chi. A fuzzy image metric with application to fractal coding. *IEEE Transactions on Image Processing*, 11(6):636–643, 2002.
- D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar. *Handbook of Fingerprint Recognition*. New York: Springer, 2003.
- E. Pekalska and Robert P. W. Duin. Dissimilarity representations allow for building good classifiers. *Pattern Recognition Letters*, 23(8), 2002.
- H. Saigo, J.-P. Vert, N. Ueda, and T. Akutsu. Protein homology detection using string alignment kernels. *Bioinformatics*, (11):1682–1689, 2004.
- R. Schapire, Y. Freund, P. Bartlett, and W. Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *Annals of Statistics*, 26:1651–1686, 1998.
- R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37:297–336, 1999.
- H. Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(2):227–244, 2000.
- P. Simard, Y. LeCun, and J. Denker. Efficient pattern recognition using a new transformation distance. In S. J. Hanson, J. D. Cowan, and C. L. Giles, editors, *Advances in Neural Information Processing Systems*, volume 5, 1993.
- N. Srebro. How good is a kernel when used as a similarity measure? In *The 20th Annual Conference on Learning Theory*, 2007.
- M. Sugiyama, M. Krauledat, and K.-R. Müller. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8:985–1005, May 2007.
- V. Vapnik. *Statistical Learning Theory*. John Wiley and Sons Inc., 1998.
- L. Wang, M. Sugiyama, C. Yang, Z. Zhou, and J. Feng. On the margin explanation of boosting algorithms. In *21st Annual Conference on Learning Theory*, 2008.
- L. Wang, Y. Zhang, and J. Feng. On the euclidean distance of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1334–1339, 2005.
- W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Computing Surveys*, 35:399–458, 2003.