On Learning with Dissimilarity Functions

Liwei Wang WANGLW@CIS.PKU.EDU.CN
Cheng Yang YANGCH@CIS.PKU.EDU.CN
Jufu Feng FJF@CIS.PKU.EDU.CN
State Key Laboratory of Machine Perception,School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, P.R.China

Abstract

We study the problem of learning a classification task in which only a dissimilarity function of the objects is accessible. That is, data are not represented by feature vectors but in terms of their pairwise dissimilarities. We investigate the sufficient conditions for dissimilarity functions to allow building accurate classifiers. Our results have the advantages that they apply to unbounded dissimilarities and are invariant to order-preserving transformations. The theory immediately suggests a learning paradigm: construct an ensemble of decision stumps each depends on a pair of examples, then find a convex combination of them to achieve a large margin. We next develop a practical algorithm called Dissimilarity based Boosting (DBoost) for learning with dissimilarity functions under the theoretical guidance. Experimental results demonstrate that DBoost compares favorably with several existing approaches on a variety of databases and under different conditions.

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 any 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 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 the 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. For some applications such as image retrieval, this representation has the advantage that it is more convenient to define a dissimilarity measure than a set of meaningful features (Jacobs et al., 2000; Jain & Zongker, 1997). In addition, dissimilarity functions can often be defined on structured objects. This procedure thus provides a bridge between classical and the structural approach to pattern classification (Graepel et al., 1999; Goldfarb, 1985).

The simplest method to classify objects in dissimilarity representations is the nearest neighbor (NN) rule. Despite of the theoretical result linking its asymptotic error rate to the Bayes optimal risk, NN suffers from a number of drawbacks like high computational complexity, sensitive to the choice of distance measure and intolerance of noisy data (Breiman et al., 1984). This is due to the fact that NN relies entirely on the local topology. If the dissimilarity measure is not perfect or there are noisy examples, the performance of NN degrades significantly.

In contrast to NN, several authors proposed more global classification algorithms. The underlying idea is that global decision rules are less sensitive to the choice of distance measure and noise. 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 (Pekalska et al., 2002; 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 SVM in the new space (Pekalska & Duin, 2002; Balcan et al., 2006; Balcan et al., 2004) . All these algorithms demonstrate superior performance to NN on a number of datasets.

More recently, Balcan and Blum (2006) developed a theory of learning with similarity functions. They defined a no-

Appearing in *Proceedings of the 24th International Conference on Machine Learning*, Corvallis, OR, 2007.

tion of what it means for a pairwise function to be a good similarity function, and give sufficient conditions for a normalized similarity function to allow one to learn well. This theory immediately suggests the algorithms of the second type described above, and therefore provides a theoretical explanation of their good empirical performances.

In this paper we study both theoretical and practical issues of learning with dissimilarity functions. We first extend Balcan and Blum (2006) theory on normalized similarity functions to unbounded dissimilarity functions. We give sufficient conditions for dissimilarity functions to allow one learn well. One advantage of our result is that our notions of good dissimilarity functions are invariant to order-preserving transformations. Interestingly, the theory suggests a learning paradigm different from all the aforementioned algorithms: Construct an ensemble of decision stumps of special forms and then find a convex combination of them to achieve a large margin. We then develop more practical algorithms under this theoretical guidance. In particular, boosting is adopted due to its ability on obtaining large margin distribution.

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 next provide experimental evidence of the benefits of our algorithm in Section 4, and conclude in Section 5.

2. Theory

2.1. Notations

By dissimilarity we mean any nonnegative two parameter 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 $z, z', z'' \cdots$, 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 $\mathrm{sgn}(x) = 1$ if $x > 0$ and -1 otherwise.

It is worth while to point out that although we focus on dissimilarity functions, extension to similarity functions is trivial. 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 obtains the theory for similarity functions. Therefore, the theory is a unified framework for learning with similarity and dissimilarity functions.

2.2. Sufficient Conditions for Learning with Dissimilarity Functions

We propose in this section a number of sufficient conditions for a dissimilarity function that are useful for learning, with the main results established in Definition 4 and Theorem 5.

We first give a notion of good dissimilarity function, 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.

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

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

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

One advantage of this definition is that strongly (ε, γ)-goodness of a dissimilarity function is invariant to order-preserving transformations. Formally,

Proposition 2 *Suppose that $d(\cdot, \cdot)$ and $D(\cdot, \cdot)$ are dissimilarity functions and $d(\cdot, \cdot)$ is strongly (ε, γ)-good. If for arbitrary x_1, x_2, x_1', x_2', $d(x_1, x_2) > d(x_1', x_2')$ iff $D(x_1, x_2) > D(x_1', x_2')$, then $D(\cdot, \cdot)$ is also strongly (ε, γ) -good. In particular, if φ is a strictly increasing function, then $D = \varphi \circ d$ is strongly (ε, γ)-good if d is.*

The proposition is immediate from Definition 1.

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

Theorem 3 *If d is a strongly (ε, γ)-good dissimilarity function, then with probability at least $1 - \delta$ over the choice of $n = (1/\gamma^2)\, ln(1/\delta)$ pairs of examples (z_i', z_i'') with labels $y_i' = 1$, $y_i'' = -1$, $i = 1, 2, \cdots, n$, the following classifier $H(x)$ has an error rate of no more than $\varepsilon + \delta$:*

$$H(x) = \mathrm{sgn}\,[\, f(x)], \; f(x) = \frac{1}{n} \sum_{i=1}^{n} \mathrm{sgn}\,[d(x, x_i'') - d(x, x_i')].$$

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

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

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

$$E\left(sgn\left[d(x,x'') - d(x,x')\right] \mid y' = y, y'' = -y\right) \geq 2\gamma.$$

Chernoff bound then implies that:

$$P\left(y \cdot \frac{1}{n}\sum_{i=1}^{n} \text{sgn}\left[d(x,x_i'') - d(x,x_i')\right] \leq 0\right) \leq e^{-2n\gamma^2}.$$

Since the above inequality holds for every $z \in M$, we can take expectation over all $z \in M$, results in that the expected error is at most $e^{-2n\gamma^2}$. Next using Markov inequality we obtain that the probability that the error rate over the set M is larger than θ is at most $e^{-2n\gamma^2}/\theta$ for arbitrary $\theta > 0$. Finally, setting $\delta = e^{-2n\gamma^2}/\theta$ and adding the ε probability of examples z not in M completes the proof. □

Although Definition 1 and its suggested algorithm are natural, the notion of strongly (ϵ, γ)-goodness is too restrictive. Similar to the argument in (Balcan & Blum, 2006), one can show that there are many examples which are strong learnable yet do not satisfy Definition 1. However, imposing appropriate weighting functions over the instances would make inequality (1) valid with respective to the new distribution and allows one to learn well in the same way as described above. The following definition is at the core of this work. In this definition we merely assume the existence of the weighting functions, which are not necessarily known a priori. It will become clear that this assumption alone is sufficient to learn an accurate classifier. Therefore it captures a broad class of dissimilarity functions.

Definition 4 *Denote by $p(x \mid y = 1)$ and $p(x \mid y = -1)$ the conditional pdf 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 pdf $\tilde{p}(x \mid y = 1)$ and $\tilde{p}(x \mid y = 1)$ such that for all $x \in X$*

$$\frac{\tilde{p}(x \mid y=1)}{p(x \mid y=1)} < \sqrt{B}, \qquad \frac{\tilde{p}(x \mid y=-1)}{p(x \mid y=-1)} < \sqrt{B}$$

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

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

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

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

It is easily seen that (ε, γ, B)-goodness is also invariant to order-preserving transformations of the dissimilarity functions.

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 a number of base classifiers.

Theorem 5 *If d is a (ε, γ, B)-good dissimilarity function, then with probability at least $1 - \delta$ over the choice of $n = 4B^2/\gamma^2\ ln(1/\delta)$ pairs of examples (z_i', z_i'') with labels $y_i' = 1, y_i'' = -1, i = 1, 2, \cdots, n$, there exists a convex combination classifier $f(x)$ of n base classifiers $h_i(x)$*

$$\begin{aligned}f(x) &= \sum_{i=1}^{n} \alpha_i h_i(x), \sum \alpha_i = 1, \alpha_i \geq 0,\\ h_i(x) &= \text{sgn}\left[d(x,x_i'') - d(x,x_i')\right].\end{aligned}$$

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

$$P\left(y \cdot f(x) \leq \gamma/B\right) \leq \varepsilon + \delta.$$

Proof: Denote

$$w_1(x) = \frac{\tilde{p}(x \mid y=1)}{p(x \mid y=1)}, \qquad w_{-1}(x) = \frac{\tilde{p}(x \mid y=-1)}{p(x \mid y=-1)},$$

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

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

Hence

$$\tilde{P}\left(d(x,x') < d(x,x'') \mid y' = y, y'' = -y\right) \geqslant 1/2 + \gamma$$

is equivalent to

$$E\left\{w_y(x')w_{-y}(x'')sgn\left[d(x,x'') - d(x,x')\right] \mid y' = y, y'' = -y\right\} \geqslant 2\gamma.$$

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

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

Let $\alpha_i = \dfrac{w_1(x_i')\, w_{-1}(x_i'')}{\sum_j w_1(x_j')\, w_{-1}(x_j'')}$, we have:

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

Since $w_1(x_i')w_{-1}(x_i'') \leqslant B$, we obtain

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

Take expectation over all $z \in M$ then use the Markov inequality as in the proof of Theorem 3, we complete the proof. □

Theorem 5 suggests that if d is a (ε, γ, B)-good dissimilarity function, in order to learn well one only needs to draw a number of pairs of examples to build an ensemble of decision stumps, then use an independent set of examples to train a large margin convex combination of the base classifiers. Note that boosting is especially suitable for computing a large margin composite classifier (Schapire & Singer, 1999). In Section 3, we will adopt boosting to develop more practical algorithms for learning with dissimilarities.

2.3. Discussion of the Sufficient Conditions

We require in our definitions of good dissimilarity functions that most of the examples (at least $1 - \varepsilon$ 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 6 *A dissimilarity function $d(x, x')$ is said to be pseudo γ-good for the learning problem, if*

$$P\left(d(x, x') < d(x, x'') \mid y' = y, y'' = -y\right) \geqslant 1/2 + \gamma. \quad (3)$$

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

One might expect that the pseudo goodness would also imply learnable, possibly in a weak sense. However, we can show 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 definitions of good dissimilarity given in Section 2.2 can not be weakened too much.

For simplicity, we assume in the next proposition that the class probability is equal, i.e. $P(y = 1) = P(y = -1) = 1/2$.

Proposition 7 *There exists a learning problem and a pseudo γ-good ($0 < \gamma < 1/4$) dissimilarity function for this problem such that even if $n \to \infty$, it is still with high probability (close to 1) over the choice of n pairs of examples (z_i', z_i'') with labels $y_i' = 1, y_i'' = -1, i = 1, 2, \cdots, n$, that the error rate of the voting classifier is higher than $1/2$, that is,*

$$P\left(y \cdot f(x) < 0\right) > 1/2,$$
$$f(x) = \frac{1}{n} \sum_{i=1}^{n} sgn\left[d(x, x_i'') - d(x, x_i')\right].$$

Proof Sketch: For a fixed example $z = (x, y)$, when $n \to \infty$, the law of large numbers gives that

$$f(x) = \frac{1}{n} \sum sgn\left[d(x, x_i'') - d(x, x_i')\right]$$

converges in probability to

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

Note further that

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

It can be shown that when n is sufficiently large, with probability close to 1, an error occurs, i.e. $y f(x) < 0$ if

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

Denote

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

For $0 < \gamma < 1/4$, it is easy to construct a distribution such that the following two inequalities hold simultaneously:

$$E\left[g(z)\right] \geqslant 1/2 + \gamma$$
$$P\left(g(z) < 1/2\right) > 1/2.$$

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 propose a practical algorithm for learning with dissimilarity functions under the previous theoretical guidance. The algorithm is essentially Dissimilarity based Boosting, and will be referred to as DBoost.

Our main theoretical result (Theorem 5) suggests that one randomly draw a large number of pairs of examples to construct base classifiers of the form:

$$h_i(x) = \begin{cases} 1 & if \quad d(x, x_i') < d(x, x_i'') \\ -1 & otherwise \end{cases}, \quad (4)$$
$$y_i' = 1, \quad y_i'' = -1$$

Then boosting can be adopted to train, by using an independent set of examples, a convex combination of these $h_i(x)$ such that the composite classifier $f(x) = \sum \alpha_i h_i(x)$ has a large margin. In practice however, only limited examples are presented, so we have to use the data effectively. In our algorithm, we use all the data as training set and consider all pairs of examples with different labels as candidates to build the stump base classifiers. Another practical

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
HAYES	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

issue is that the base classifier (4) is too weak, we therefore strengthen its discrimination ability. These modifications are described in detail as follows.

Consider the base classifier given in (4). This is a decision stump which partitions the instance space X into two subspaces X^+ and X^- defined as:

$$\begin{aligned} X^+ &= \{x \mid d(x,x') - d(x,x'') < 0\}, \\ X^- &= \{x \mid d(x,x') - d(x,x'') \geqslant 0\}. \\ y' &= 1, \quad y'' = -1 \end{aligned}$$

If the instance space X is a Euclidean space, the decision surface of the decision stump is a hyperplane orthogonal to the line joining x', x'' and bisects the line.

We suggest using as the base classifier the following decision stump, which partitions X into X^+ and X^- as:

$$\begin{aligned} X^+ &= \left\{x \mid d^2(x,x') - d^2(x,x'') < v\right\}, \\ X^- &= \left\{x \mid d^2(x,x') - d^2(x,x'') \geqslant v\right\}, \end{aligned} \tag{5}$$

where v is a threshold introduced as a free parameter to increase the discrimination ability of the base learner. If X is a Euclidean space, the decision surface of the above decision stump is also a hyperplane orthogonal to the line passing through $x'$and x'' but biased at v. Hence (5) is a straightforward generalization of (4).

The most important issue for the design of our algorithm is how to train the decision stump under the boosting framework. In the traditional feature representation, boosting decision stump weak learners has been extensively studied. A stump is trained by exhaustively searching over all candidates and selecting the one with minimum loss with respect to the distribution of the current round (This algorithm is called FindAttrTest (Freund & Schapire, 1996)). In other words, all attributes and all possible thresholds are considered. This method however, can not be used in our dissimilarity setting without modifications. Because the implementation here needs $O(n^3)$ time, where n is the number of training examples. This is intractable even for small size tasks.

To reduce the computational cost yet maintain as much information as possible, we suggest the following strategy.

Strategy I: At round t of AdaBoost, we randomly select N pairs of training examples with different labels. The selection of each example is according to the current distribution. Therefore, examples that are harder to classify have higher probability to be selected. As t increases, the algorithm concentrates on the "boundary" data, since they are most difficult to recognize. The base classifier at each round is obtained by selecting the best one among all decision stumps determined by the N pairs of training examples. The search consumes only $O(N \cdot n)$ time.

We next consider the problem of the storage cost. It is easy to see that the storage is mainly determined by the number of *distinct* examples selected by the training algorithm. Note that the computational time for recognizing a new example also depends on the number of distinct examples, since we need to compute the dissimilarities between the new instance and each of these examples. If boosting runs a large number of rounds, this number may become large. To control it we suggest an alternative strategy for training the base classifier.

Strategy II: We construct a prototype set S. Initially S is empty. At each round of the boosting, we count the number of elements in S. If it is less than a predefined number C, we train the base classifier using Strategy I, and then put the two prototypes selected by the base learner in S. If at some round, S has already contained more elements than C, then we generate N pairs of training examples with different labels, and each example is chosen randomly within S according to their current weights. The base classifier is obtained by choosing the best decision stump among those determined by these N pairs of examples.

Table 2. Comparison of the performances of the five algorithms on UCI datasets.

DATA SET	NN	RLNC	RQNC	LSVM	DBOOST
BALANCE	21.4 ± 3.4	11.5 ± 2.2	13.7 ± 2.8	8.3 ± 3.6	**2.3 ± 1.9**
BREAST	4.2 ± 1.6	**2.9 ± 1.6**	2.9 ± 1.9	3.8 ± 2.7	2.9 ± 1.8
CLEVELAND	41.3 ± 6.0	33.0 ± 4.4	38.4 ± 4.5	33.6 ± 5.4	**20.5 ± 6.5**
DIABETES	32.0 ± 3.1	26.9 ± 1.7	28.4 ± 3.6	**24.3 ± 2.7**	24.7 ± 2.4
ECHO	14.2 ± 7.3	9.5 ± 6.0	14.2 ± 6.7	**9.5 ± 5.8**	17.1 ± 8.1
GERMAN	34.2 ± 3.2	**25.6 ± 1.9**	30.5 ± 1.3	26.5 ± 3.1	26.8 ± 2.4
HAYES	30.6 ± 8.2	18.1 ± 5.3	21.9 ± 4.4	**16.8 ± 3.6**	24.4 ± 7.1
HEPATITIS	19.7 ± 7.0	20.3 ± 7.0	18.3 ± 0.8	19.0 ± 6.6	**16.3 ± 8.4**
IMAGE	3.9 ± 0.8	5.2 ± 0.4	4.0 ± 3.3	5.1 ± 0.6	**2.3 ± 0.5**
IONOSPHERE	13.7 ± 4.2	7.7 ± 3.2	6.0 ± 1.3	**5.1 ± 3.2**	6.8 ± 1.2
IRIS	**4.4 ± 3.0**	4.7 ± 1.6	6.0 ± 0.3	4.6 ± 3.0	6.0 ± 4.9
LETTER	4.3 ± 0.3	23.0 ± 0.4	8.4 ± 8.5	6.1 ± 0.2	**2.5 ± 0.2**
LIVER	38.5 ± 5.0	34.0 ± 8.2	40.0 ± 3.8	**29.5 ± 10.2**	30.1 ± 4.5
MONK1	13.8 ± 3.3	23.4 ± 2.2	14.9 ± 4.0	0.7 ± 1.2	**0.0 ± 0.0**
MONK2	20.8 ± 4.2	34.7 ± 1.6	7.5 ± 0.6	15.6 ± 4.3	**2.3 ± 2.0**
MONK3	11.1 ± 2.6	3.6 ± 0.8	3.8 ± 1.2	**1.4 ± 1.0**	4.2 ± 1.7
SATIMAGE	9.5 ± 0.8	9.4 ± 0.6	11.1 ± 1.8	9.5 ± 1.3	**7.3 ± 1.3**
VEHICLE	34.9 ± 3.0	26.6 ± 3.2	43.9 ± 0.4	**23.4 ± 1.4**	25.2 ± 1.3
VOWEL	**1.5 ± 0.9**	9.1 ± 1.8	6.1 ± 17.4	3.4 ± 0.7	2.8 ± 0.8
WDBC	8.4 ± 2.4	6.1 ± 2.5	26.1 ± 8.7	6.8 ± 1.9	**2.5 ± 1.7**
WINE	26.5 ± 7.4	27.5 ± 4.3	28.6 ± 4.7	26.8 ± 1.4	**2.2 ± 2.3**
WPBC	35.6 ± 5.7	29.4 ± 5.2	26.3 ± 3.9	**23.7 ± 5.7**	25.3 ± 6.9
AVERAGE	19.3	17.8	18.2	13.8	11.6
#WINS	2	2	0	8	10

Using this strategy, the algorithm selects at most $C+1$ prototypes.

In our implementation, the base classifiers are constructed and combined by the real version AdaBoost algorithm (Schapire & Singer, 1999), for which the decision stumps output unbounded real values. Although our theory is for binary classification, the DBoost algorithm applies to multiclass problems as well by simply using the multiclass boosting algorithm AdaBoost.MH as the booster.

4. Experiments

In this section we evaluate our DBoost algorithm and compare to several existing approaches for learning with dissimilarity functions. In all the experiments we use Strategy I to train the base classifiers. Because of the space, we cannot provide the detailed data of Strategy II, which usually results in minor deficit in accuracy while the storage and recognition time can be saved by $2/3$. Finally, the number of pairs of examples selected at each round of DBoost is set to be 100.

The approaches we compare with are the nearest neighbor (NN) rule, the RLNC/RQNC method (Pekalska & Duin, 2005) and the linear SVM (LSVM) algorithm (Balcan & Blum, 2006). We give a brief description as follows.

RNLC/RNQC (Pekalska & Duin, 2005): This method represents each object x as a vector via its dissimilarities to a set of prototypes $p_1, p_2, \cdots, p_m$. This can be viewed as a mapping Ψ from the instance to a m -dimensional space.

$$\Psi : x \rightarrow (d(x, p_1), d(x, p_2), \cdots, d(x, p_m))$$

Then regularized linear/quadratic normal density-based classifier (Ripley, 1996) is employed to classify the data.

LSVM (Balcan & Blum, 2006): This algorithm is suggested by the theory in the same paper. After a mapping like the one described above but via normalized similarity functions, linear SVM is adopted to find a large margin separator. In our experiment, the normalized similarity function K is obtained by transforming the dissimilarity function d as $K = \exp(-d^2/\sigma)$. We use libsvm (Chang & Lin, 2001) for the SVM implementation. The parameters σ and m (number of prototypes selected) are tuned on the training set.

The first set of experiments is conducted on 22 benchmark datasets from UCI repository. Each dataset is used in a five-fold cross validation fashion. The datasets are described in Table 1. Although in these datasets objects are represented by feature vectors, we feed the algorithms only the Euclidean distances between the data.

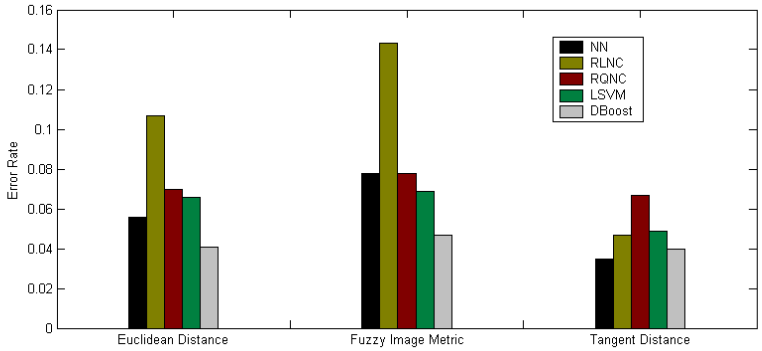

Figure 1. Comparison of the algorithms on USPS dataset using three dissimilarity functions.

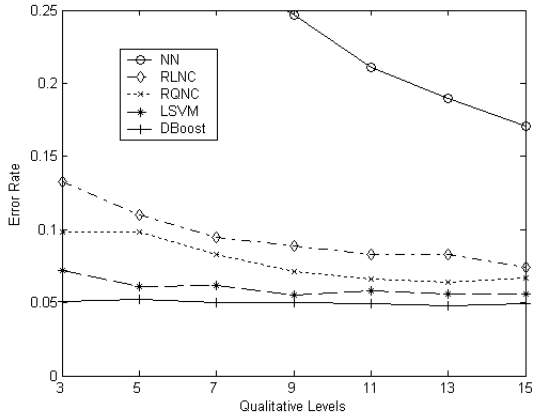

Figure 2. Comparison of the algorithms using qualitative dissimilarity functions on the USPS database.

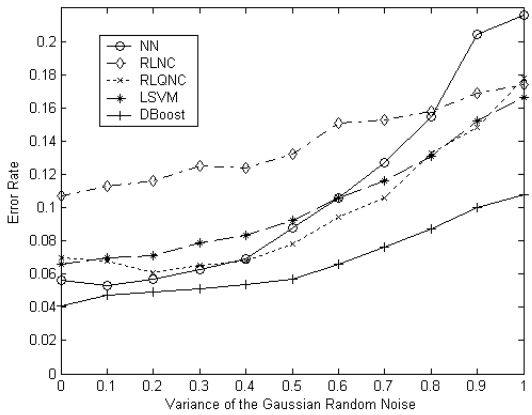

Figure 3. Comparison of the algorithms on the USPS database with noise.

The goal of this experiment is to compare the algorithms on a variety of domains. The results are shown in Table 2. DBoost outperforms all the other approaches both in terms of the average error rate of the 22 databases and the number of datasets on which the algorithm achieves the best performance.

In the next experiment we focus on image classification. As mentioned earlier, it is more convenient to directly define dissimilarities between images than to construct meaningful features. Many dissimilarity measures of images have been proposed in literature. We adopt in this experiment three measures: Euclidean distance, Fuzzy Image Metric (Li et al., 2002), and Tangent distance (Simard et al., 1993). The aim of this experiment is to compare the algorithms using different dissimilarities. We perform the experiment on the USPS database which consists of images of handwritten digits. The dataset has been partitioned into a fixed training set and testing set, consisting of 7291 and 2007 examples respectively. The results are depicted in Fig1. DBoost has the best performance on the Euclidean distance and the Fuzzy Image Metric. With the tangent distance which intellectually incorporates strong domain specific knowledge and is believed to be a perfect distance for this task, DBoost is not as good as NN but still outperforms the other three algorithms.

We then consider the situation in which the dissimilarity is far from accurate. For instance, when people make subjective evaluation of the similarity between images, only qualitative (discrete) values can be given. As an example, similar, average and dissimilar are possible values of a three-level qualitative dissimilarity. We compare the performance of our algorithm to others 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 Fig2. DBoost outperforms the other algorithms consistently on all the qualitative levels. Observe that DBoost is the most insensitive to the choice dissimilarity measures. Even with the 3-level measures, for which the information of local topology has been mostly lost, DBoost still has low error rate.

In the final experiment we evaluate the performance of our

algorithm against noisy data. We add to the USPS images Gaussian white noise with different Energies, and feed the algorithms with the Euclidean distances. The results are depicted in Fig3. DBoost is relatively robust to noise.

5. Conclusion and Discussion

In this contribution we have given sufficient conditions for dissimilarity functions to allow one to learn well. Our definition of good dissimilarities applies to unbounded functions and is invariant to order-preserving transformations. The theory immediately suggests a simple algorithm: using boosting to construct and combine a large number of base classifiers each depends on a pair of examples. We then develop a more practical algorithm DBoost by generalizing the base learner and suggesting strategies for training base classifiers to make the computation tractable. Our approach compares favorably with several existing algorithms on a variety of databases.

Although we use stumps as the base classifiers in our algorithm, one can use instead decision trees by further splitting each node as in (5). Incidentally, this splitting approach for decision trees was previously suggested by Hinton and Revow (1996) . It is expected that the use of decision tree as base classifiers would bring additional benefits.

Acknowledgments

This work was supported by NSFC(60575002,60635030), NKBRPC(2004CB318000) and Program for New Century Excellent Talents in University. The authors would like to thank Wenquan Xu for useful comments.

References

Balcan, M.-F., & Blum, A. (2006). On a theory of learning with similarity functions. *International Conference on Machine Learning*.

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

Balcan, M.-F., Blum, A., & Vempala, S. (2006). Kernels as features: On kernels, margins, and low-dimensional mappings. *Machine Learning*, *65*, 79–94.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees*. Wadsworth.

Chang, C. C., & Lin, C. J. (2001). *A library for support vector machines*. Available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

Freund, Y., & Schapire, R. E. (1996). Experiments with a new boosting algorithm. *International Conference on Machine Learning*.

Goldfarb, L. (1985). A new approach to pattern recognition. *In L.N. Kannal and A. Rosenfeld (Ed.), Progress in Pattern Recognition*, *2*, 241–402.

Graepel, T., Herbrich, R., Bollmann-sdorra, P., & Obermayer, K. (1999). Classification on pairwise proximity data. *Advances in Neural Information Processing Systems*.

Hinton, G. E., & Revow, M. (1996). Using pairs of datapoints to define splits for decision trees. *Advances in Neural Information Processing Systems*.

Jacobs, D. W., Weinshall, D., & Gdalyahu, Y. (2000). Classification with nonmetric distances: Image retrieval and class representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*, 583–560.

Jain, A. K., & Zongker, D. E. (1997). Representation and recognition of handwritten digits using deformable templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *19*, 1386–1391.

Li, J., Chen, G., & Chi, Z. (2002). A fuzzy image metric with application to fractal coding. *IEEE Transactions on Image Processing*, *11*, 636–643.

Maltoni, D., Maio, D., Jain, A. K., & Prabhakar, S. (2003). *Handbook of fingerprint recognition*. New York: Springer.

Pekalska, E., & Duin, R. P. W. (2002). Dissimilarity representations allow for building good classifiers. *Pattern Recognition Letters*, *23*, 943–956.

Pekalska, E., & Duin, R. P. W. (2005). *The dissimilarity representation for pattern recognition*. World Scientific.

Pekalska, E., Paclík, P., & Duin, R. P. W. (2002). A generalized kernel approach to dissimilarity-based classification. *Journal of Machine Learning Research*, *2*, 175–211.

Ripley, B. (1996). *Pattern recognition and neural networks*. Cambridge: Cambridge University Press.

Schapire, R. E., & Singer, Y. (1999). Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, *37*, 297–336.

Simard, P., Cun, Y. L., & Denker, J. (1993). Efficient pattern recognition using a new transformation distance. *Advances in Neural Information Processing Systems*.

Zhao, W., Chellappa, R., Phillips, P. J., & Rosenfeld, A. (2003). Face recognition: A literature survey. *ACM Computing Surveys*, *35*, 399–458.