Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Margin distribution explanation on metric learning for nearest neighbor classification

Peng-Cheng Zou*, Jiandong Wang, Songcan Chen, Haiyan Chen

College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China

ARTICLE INFO

Article history:
Received 18 June 2015
Received in revised form
4 November 2015
Accepted 12 November 2015
Communicated by Shiliang Sun

Keywords:
Distance metric learning
Nearest neighbor classification
Margin distribution
Generalization performance

ABSTRACT

The importance of metrics in machine learning and pattern recognition algorithms has led to an increasing interest for optimizing distance metrics in recent years. Most of the state-of-the-art methods focus on learning Mahalanobis distances and the learned metrics are in turn heavily used for the nearest neighbor-based classification (NN). However, until now no theoretical link has been established between the learned metrics and their performance in NN. Although some existing methods such as large-margin nearest neighbor (LMNN), have employed the concept of large margin to learn a data-dependent metric, the link between the margin and the generalization performance for the metric is not fully understood. Though the recent work has indeed provided tenable margin distribution explanation on Boosting, the margin used in metric learning is quite different from that in Boosting. Thus, in this paper we try to analyze the effectiveness of metric learning algorithms for NN from the perspective of the margin distribution and provide a general and effective evaluation criterion for metric learning. On the one hand, we derive the generalization error upper bound for NN with respect to the Mahalanobis metric. On the other hand, the experiments on several benchmark datasets using existing metric learning algorithms demonstrate that large margin distribution can be obtained by these algorithms. Motivated by our analysis above, we also present a novel margin based metric learning algorithm for NN, which explicitly enlarges the margin distribution on various datasets and achieves very competitive results with the existing metric learning algorithms.

1. Introduction

Distance metric learning is a fundamental problem in machine learning and pattern recognition. It is critical to many machine learning algorithms, such as nearest neighbor classification, support vector machine and K-means [1,2]. A number of recent algorithms have been proposed for distance metric learning. Generally, these approaches are based on the reasonable intuition that a good metric should make the distances between similar data points smaller, while the distances between dissimilar data points larger. Among these approaches, Mahalanobis distance metric learning [3–6] is a well-studied and successful framework, whose goal is to learn a positive semi-definite (PSD) matrix to cater for the above-mentioned intuition by linearly projecting the data into a new feature space where the standard Euclidean distance can desirably be performed. Then the distance metric learned can also further be used in subsequent learning tasks such as classification and rank.

Although existing metric learning algorithms have been shown to be useful in many real-world applications [7–9], more theoretical understanding is still needed for metric learning. One important problem is the generalization guarantee for metric learning, i.e., bounding its performance on unseen data from its performance on the training examples. However, establishing a generalization guarantee for the learned metric is still challenging and only has been investigated recently from a theoretical standpoint [1,10]. The problem is currently studied in the two facets: the first one is the consistency of the learned metric, which tries to bound the deviation between the empirical performance of the metric on the training samples and its generalization performance on unseen data, and the second one is to understand the link between the learned metric and the generalization performance of the subsequent learning algorithms with this metric, with such a link, we can get some insights to design new learning algorithms.

Several recent works have proposed for the first facet by studying the convergence of the empirical risk of metric learning to the true risk over the unknown probability distribution. For example, Jin et al. [11] use the notion of uniform stability adapted to metric learning with Frobenius norm regularization to derive generalization bounds; Bellet and Habrard [12] adapt the notion of

* Corresponding author.
E-mail address: zou_pc@163.com (P.-C. Zou).

http://dx.doi.org/10.1016/j.neucom.2015.11.016
0925-2312/© 2015 Elsevier B.V. All rights reserved.

algorithmic robustness to the metric learning setting to derive generalization bounds for any matrix norm as regularizer. Cao et al. [13] apply the notion of Rademacher complexity to derive bounds for metric learning. Meanwhile, the results of the second facet of generalization have only been obtained for linear classification so far. For example, Bellet et al. [14] resort to the theory of learning with $(\varepsilon, \gamma, \tau)$-good similarity function and explore the link between properties of a similarity function and the generalization of a linear classifier built from this similarity. Guo and Ying [15] extend the results of Bellet et al. [14] to the cases based on other several matrix norms using a Rademacher complexity analysis.

In this paper, we focus on the second facet of generalization for metric learning. It is noted that, the learned metrics are also heavily used for other learning algorithms, especially the nearest neighbor classification which is a local nonlinear classifier rather than the linear classifiers as those studied in this facet. In general, the Mahalanobis metrics learned from training pairs or triplets are plugged in a k-NN classifier and often lead to greater accuracy than the standard Euclidean distance [11], but no direct theoretical evidence supports this behavior. Although some existing methods such as LMNN [4], have employed the concept of large margin to design the algorithms and many empirical evidences have also demonstrated that these methods generally perform better than other metric learning algorithms such as neighborhood component analysis (NCA) or relevant component analysis (RCA) for NN [4,6,16], the link between the margin with respect to the learned metric and the generalization performance for subsequent nearest neighbor classification is not fully understood yet. Besides, inspired by the recent theoretical results that the margin distribution rather than a single margin is really crucial for the generalization performance [17], we also try to analyze the link between the learned Mahalanobis metric and the performance of nearest neighbor classification from the perspective of margin distribution and provide a general and effective evaluation criteria for metric learning. Different from the situation of the Boosting or support vector machine (SVM) analyzed by [17] where the margin is defined specifically for the Boosting or SVM, our margin is defined not for the specific metric learning algorithm but for the subsequent nearest neighbor classification with the learned metric. Thus, our learning situation is more complex in that the metric learning is usually independent of classification of the target classifiers. In addition, due to the fact that the target classifier like the NN is a lazy learner which does not involve training, thus it becomes relatively reasonable to select it as a criterion of evaluating the performance of learned metric induced from the nearest neighbor classification.

Throughout this paper we will use the 1-NN as a target classifier due to its excellent performance with the error rate less than a factor of 2 Bayes error rate [18]. In our analysis, we first extend the margin for 1-NN defined in [19] with the Mahalanobis metric and then prove a generalization error bound of 1-NN on a Mahalanobis metric, the resulting bound guarantees good performance for the learned metric while keeping the margin distribution large. Next, we provide more theoretical evidence for the existing large margin metric learning algorithms and explain the reason why the metric learned from training triplets can be useful when used in conjunction with the nearest-neighbor classifier. Actually, by overcoming the scale ambiguity of the margin of metric learning, our study further enriches the results in [20] which is specific to feature selection. Therefore, we draw a more general conclusion that is applicable for any linear transformation of the original feature space including Mahalanobis metric learning and other feature learning methods, as a result, motivating us to design novel metric learning algorithms based on the large margin distribution criterion.

The rest of this paper is organized as follows. In Section 2, we discuss the margins in machine learning community and present the margin with respect to the Mahalanobis metric. In Section 3, we present a theoretical generalization analysis to explain the link between the margin distribution with respect to the learned metric and the performance of 1-NN. Empirical evidence on several benchmark datasets with the representative metric learning algorithms is provided to verify the theoretical analysis and explain the effectiveness of these existing methods in Section 4. We provide more empirical evidences by presenting a novel metric learning algorithm for NN which directly optimizes the margin in Section 5. Conclusion is given in Section 6.

2. Margin for 1-NN with the learned metric

A margin [21,22] is essentially a geometric measure for evaluating the confidence of a classifier when making decision. It is used both for theoretic generalization bounds and as guideline for designing algorithms. For example, it is well known that the margin is a fundamental issue of SVMs, and recently the margin theory for Boosting has also been demonstrated, establishing a connection between these two mainstream approaches [17]. The recent theoretical results disclosed that the margin distribution rather than a single margin is really crucial for the generalization performance [23]. In the specific context of metric learning, many algorithms such as LMNN are also designed with the concept of large margin with the goal that the k-nearest neighbors always belong to the same class while examples from different classes are separated. Besides, the margin of 1-NN has also been proposed and is used for feature selection [20]. Inspired by these previous works, we first extend the margin for 1-NN with the Mahalanobis metric such that we can analyze the link between the learned metric and the performance of nearest neighbor classification from the perspective of margin distribution.

The first focus of attention is on the definition of margin for nearest neighbor classification with respect to the Mahalanobis metric. Generally, there are two types of margins. The first type is sample-margin, which measures the distance between the data point and the decision boundary induced by the classifier, such as the margin in SVM [24]. The other one is hypothesis-margin, which measures the distance between the hypothesis and the closest hypothesis that assigns alternative label to the given data point. Boosting algorithms usually use this type of margin as the distance among different hypotheses. In the context of 1-NN, both types of margins can be found [25]. The hypothesis-margin is half the difference between the distance to the *nearmiss* and the distance to the *nearhit*, and can be easily computed as follows:

$$\rho_S(x) = \frac{1}{2}\left(\|x - nearmiss(x)\| - \|x - nearhit(x)\|\right) \tag{1}$$

where ρ_S denotes the hypothesis-margin of the new instance x with respect to the set of training samples S, $\|\cdot\|$ is the Euclidean distance, *nearmiss*(x) and *nearhit*(x) denote the nearest point to x in S with the different and the same label, respectively. In [19], the authors proved that the hypothesis-margin of 1-NN lower bounds the sample margin, that is, large hypothesis-margin for 1-NN ensures large sample-margin. Therefore, we can only use the hypothesis-margin to define the margin of 1-NN on the Mahalanobis metric.

In the specific context of Mahalanobis metric learning, the distance function between the training example x_i and the x_j is $d_M(x_i, x_j) = \|x_i - x_j\|_M = \sqrt{(x_i - x_j)^T M (x_i - x_j)}$. It is parameterized by a matrix M, which is usually a positive semi-definite matrix, and can be further decomposed as $M = L^T L (L \in \mathcal{R}^{N \times r}, r \le N)$. Thus, we can formulate the hypothesis-margin of 1-NN as the function of the learned metric M.

Definition 1. Let S be a set of samples and x be an instance. Let M be a positive semi-definite matrix, $M = L^T L$, then the margin of x is

$$\rho_S^M(x) = \frac{1}{2}(\|x - nearmiss(x)\|_M - \|x - nearhit(x)\|_M) \quad \text{or}$$

$$\rho_S^L(x) = \frac{1}{2}(\|Lx - L \cdot nearmiss(x)\| - \|Lx - L \cdot nearhit(x)\|) \tag{2}$$

Definition 1 naturally extends the original hypothesis-margin of 1-NN, so that the learned metric will affect the margin through the distance measure. Since an arbitrarily large margin can be given by enlarging its scale, such as $\rho_S^{(\lambda \cdot M)}(x) = \sqrt{\lambda}\rho_S^M(x)$, we introduce a normalization $Tr(M) = 1$ to guarantee that

$$\|x_i|_M = \|Lx_i\| \le \|x_i\|$$

which can be proved as follows:

Proof. $\|\cdot\|_F$ denotes the Frobenius norm and $\|L\|_F = \sqrt{\mathrm{Tr}(L^T L)}$. Thus, $\|L\|_F \cdot \|x_i\|_F = \sqrt{\mathrm{Tr}(L^T L)} \cdot \|x_i\| = \sqrt{\mathrm{Tr}(M)} \cdot \|x_i\|$. When $\mathrm{Tr}(M) = 1$, it follows that $\|L\|_F \cdot \|x_i\|_F = \|x_i\|$. Since Frobenius norm satisfies the consistency condition, we have $\|Lx_i\|_F \le \|L\|_F \cdot \|x_i\|_F$. Therefore, $\|Lx_i\|_F = \|Lx_i\| \le \|x_i\|$.□

Thus the scale ambiguity of the margin with the learned metric can be removed.

3. Theoretical analysis

In this section we use the margin defined with the learned metric to prove a generalization error bound for 1-Nearest Neighbor which is used both feature selection and large margin principle to prove a finite sample generalization bound for 1-NN. However, to the best of our knowledge, the margin of 1-NN has not been investigated when a Mahalanobis metric learned from a set of training pairs or triplets being plugged into 1-NN. Here, we first provide a further result which enriches the results specific to feature selection in [20]: a large margin led by an effective metric (any linear transformation of the original feature space) provides enough evidence to obtain a useful bound on the generalization error for 1-NN. If the metric learning algorithm leads to a large margin, the bound guarantees it generalizes well.

In order to prove our conclusion, we use the following notation:

Definition 2. Let D be a distribution over $\mathcal{X} \times \{\pm 1\}$ and $h : \mathcal{X} \longrightarrow \{\pm 1\}$ a classification function. We denote the generalization error of h with respect to D by $err_D(h)$:

$$err_D(h) = \Pr_{x,y \sim D}[h(x) \ne y]$$

For a set of samples $S = \{(x_k, y_k)\}_{k=1}^m \in (X \times \{\pm 1\})^m$ and a constant $\gamma > 0$, we define the $\gamma - sensitive$ training error to be

$$\hat{err}_S^\gamma(h) = \frac{1}{m}|\{(k : h(x_k) \ne y_k) \text{or} [x_k \ has\ sample-margin < \gamma)\}|$$

Our main result is in the following theorem.

Theorem 1. *Let D be a distribution over $\mathcal{R} \times (\pm 1)$ which is supported on a ball of radius R in $\mathcal{R}^N$. Let $\delta > 0$ and let S be a set of samples of size m such that $S \sim D^m$. With probability $1 - \delta$ over the random choice of S, for any set of metric M and any $\gamma \in (0, 1]$*

$$err_D(h) \le \hat{err}_S^\gamma(h) + \sqrt{\frac{2}{m}\left(d \ln\left(\frac{34em}{d}\right)\right) \log_2(578m) + \ln\left(\frac{8}{\gamma\delta}\right)}$$

where h is the 1-NN classifier when the distance is measured on the metric $M(M = L^T L)$ and $d = \left(64R/.\gamma\right)^{\dim(Lx)}$. Moreover, $\dim(Lx)$ denotes the dimension of the feature space projected by L, and e is natural constant. Note that the theorem holds when sample-margin is replaced by hypothesis-margin since the latter is the lower bound of the former.

In order to prove Theorem 1, we begin by proving a basic lemma which shows that the class of the nearest neighbor classifiers with respect to a Mahalanobis metric M is a subset of the class of 1-Lipschitz functions. Let $NN_M^S(\cdot)$ be a function, where the sign of $NN_M^S(x)$ is the label that the nearest neighbor rule assigns to x, and the magnitude is the sample-margin.

Lemma 1. *Let M be a positive semi-definite metric matrix $M = L^T L$, and S be a set of labeled samples. Then, for any $x_1, x_2 \in \mathcal{R}^N$:*

$$\left|NN_M^S(x_1) - NN_M^S(x_2)\right| \le \|Lx_1 - Lx_2\|$$

Proof. Let $x_1, x_2 \in X$. If $NN_M^S(x_1)$ and $NN_M^S(x_2)$ have the same sign, let $z_1, z_2 \in \mathcal{R}^{\dim(Lx)}$ be the points on the decision boundary of the 1-NN which are closest to Lx_1 and Lx_2 respectively. From the definition of z_1, x_2, it follows that $NN_M^S(x_1) = \|Lx_1 - z_1\|$ and $NN_M^S(x_2) = \|Lx_2 - z_2\|$. Thus,

$$\begin{aligned} NN_M^S(x_2) &= \|Lx_2 - z_2\| \le \|Lx_2 - z_1\| \le \|Lx_2 - Lx_1\| + \|Lx_1 - z_1\| \\ &= \|Lx_2 - Lx_1\| + NN_M^S(x_1). \end{aligned}$$

Similarly, we have

$$NN_M^S(x_1) \le \|Lx_2 - Lx_1\| + NN_M^S(x_2).$$

Then, we obtain that

$$\left|NN_M^S(x_1) - NN_M^S(x_2)\right| \le 1 \cdot \|Lx_1 - Lx_2\|.$$

If $NN_M^S(x_1)$ and $NN_M^S(x_2)$ have different signs, as $NN_M^S(\cdot)$ is continuous, there must be a point z on the line that connects Lx_1 and Lx_2. Let z be the crossing point between this line and the decision boundary. Thus,

$$\left|NN_M^S(x_1)\right| = \|Lx_1 - z_1\| \le \|Lx_1 - z\|$$
$$\left|NN_M^S(x_2)\right| = \|Lx_2 - z_2\| \le \|Lx_2 - z\|.$$

Then we obtain

$$\left|NN_M^S(x_1) - NN_M^S(x_2)\right| = \left|NN_M^S(x_1)\right| + \left|NN_M^S(x_2)\right|$$

Table 1
Benchmark datasets used in our experiments.

Dataset	# of classes	# of feature	# of training	# of test
SPECTF Heart	2	44	187	80
sonar	2	60	146	62
ILPD	2	10	409	174
diabetes	2	8	538	230
cardiotocography	10	21	1489	637
segmentation	7	19	1617	693
satellite	6	36	4505	1930
letter	26	16	14,000	6000
dna	3	180	1811	775
usps	10	50	6209	2789
mnist	10	50	60,000	10,000
protein	3	50	15062	6454

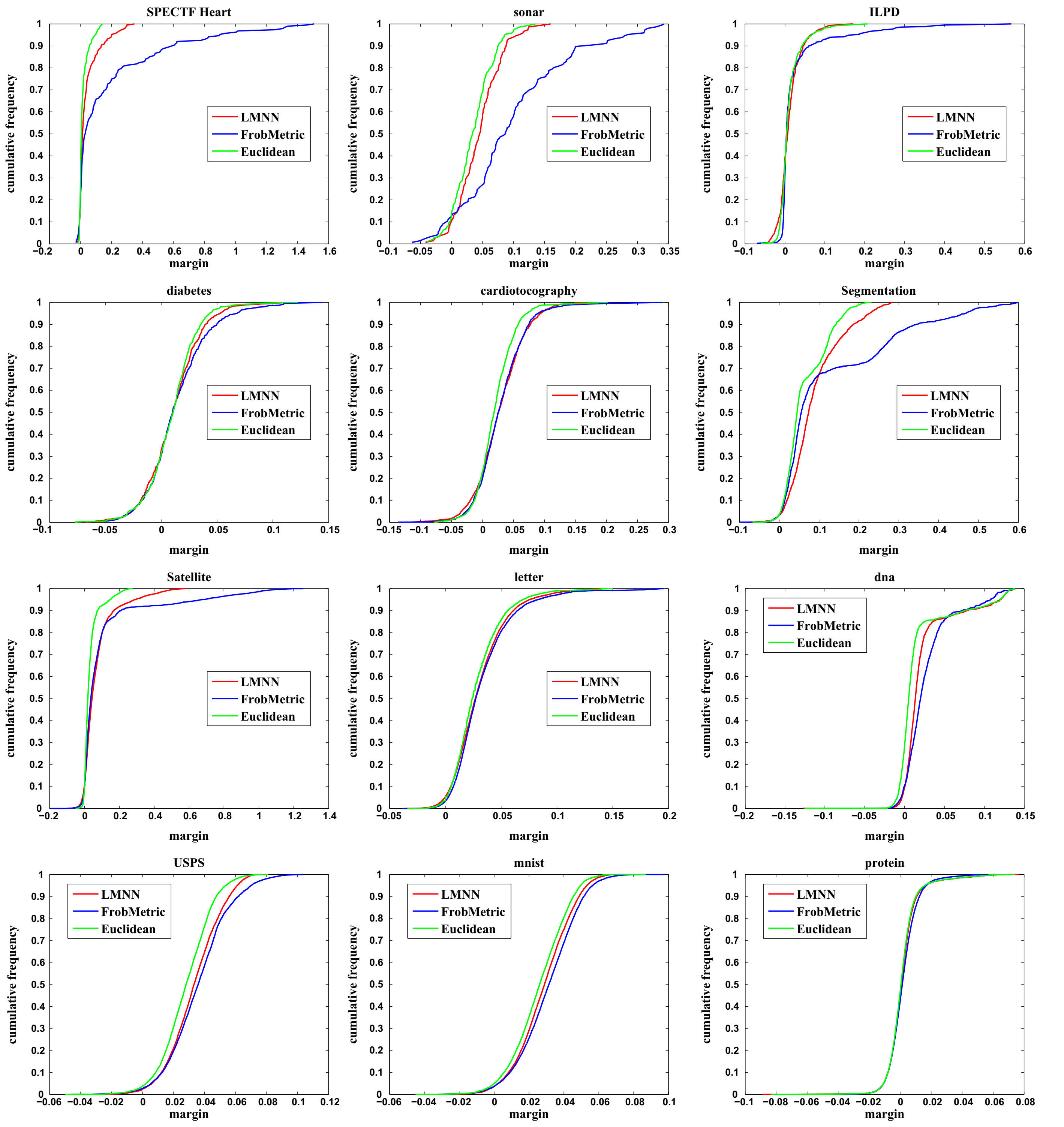


Fig. 1. Cumulative frequency (y-axis) with respect to the margin (x-axis) of Euclidean, LMNN and FrobMetric on all the datasets. The more right the curve, the larger the accumulated margin.

$$\leq \|Lx_1 - z\| + \|Lx_2 - z\| = 1 \cdot \|Lx_1 - Lx_2\|.$$

This completes the proof.□

Then, the main theoretical tool for proving Theorem 1 is the following:

Theorem 2 (*Bartlett [26]*). *Let H be a class of real valued functions. Let S be a set of samples of size m generated i.i.d. from a distribution D over $\mathcal{X} \in \{\pm 1\}$ then with probability $1-\delta$ over the choices of S, every $h \in H$ and every $\gamma \in (0, 1]$, let $d = fat_H(\gamma/.32)$:*

$$err_D(h) \leq \hat{er}_S^{\gamma}(h) + \sqrt{\frac{2}{m}\left(d \ln\left(\frac{34em}{d}\right)\right) \log_2(578m) + \ln\left(\frac{8}{\gamma\delta}\right)}$$

Next we turn to prove Theorem 1.

Proof. Let M be a positive semi-definite metric matrix $M = L^T L$ and $\gamma > 0$. In order to use Theorem 2, we only need to compute the fat-shattering dimension of 1-NN with the metric M. According to Lemma 1, the class of 1-NN classifiers on the learned metric is a subset of the class of 1-Lipschitz functions. Thus, we can bound the fat-shattering dimension of 1-NN by the dimension of Lipschitz functions. Since D is supported in a ball of radius R, and $\|Lx\| \le \|x\|$ has been proved, we only need to calculate the fat-shattering dimension of Lipschitz functions acting on points in $\mathcal{R}^{\dim(Lx)}$ with the norm bounded by R. According to Theorem 1 in [20], the $fat_\gamma - dimension$ of the 1-NN functions on the learned metric M is thus bounded by the largest γ packing of a ball in $\mathcal{R}^{\dim(Lx)}$ with radius R, which in turn is bounded by $(2R/\gamma)^{\dim(Lx)}$. Therefore, given a fixed metric matrix M, we can apply to Theorem 2 and use the bound on the fat-shattering dimension just calculated. As a result, we obtain the result in Theorem 1.□

From the generalization error bound in Theorem 1, it can be learned that the generalization performance of 1-NN on the learned metric is mainly influenced by the following factors: the margin distribution of 1-NN on the training set (i.e., $(1/m)|(x_k \text{ has sample} - margin < \gamma)|$), the number of training examples m and $fat_\gamma - dimension$ of the 1-NN with respect to γ. Therefore, according to Theorem 1, it can be inferred that when m and γ are fixed, the more the training examples with large margins are, the smaller the value $(1/m)|(x_k \text{ has sample} - margin < \gamma)|$ is, thus, the smaller the generalization error of 1-NN on the learned metric is. Actually, this is a general conclusion which is suitable for any linear transformations of the original feature space, naturally including those derived from not only the Mahalanobis metric learning but also other feature learning methods.

4. Empirical evidence with existing metric learning algorithms

In this section, in order to understand further the link between the learned metric and the performance of nearest neighbor classification from the perspective of margin distribution, the experiments on several benchmark datasets using existing metric learning algorithms are carried out. Firstly, to demonstrate the effectiveness of the margin distribution for the existing metric learning algorithms, we investigate the margin distribution with the learned metrics and compare the generalization performances of nearest neighbor classification yielded by the learned metrics. Secondly, for many metric learning algorithms, the triplet constraints can be constructed with different strategies, such as the NN selection or the random selection strategies [27]. By investigating the margin distribution resulted from the metrics learned from the constraints constructed with different strategies, we can explicitly explain the reason why the NN selection strategy is a better choice than the random selection strategy to construct the training constraints.

Table 1 summarizes the characteristics of 12 data sets used in our experiments, Datasets SPECTF Heart, ILPD, sonar, diabetes, cardiotocography, segmentation, satellite and letter are downloaded from the UCI repository [28], and dna, usps, mnist and protein from LIBSVM [29]. Due to the limitation of the memory of our computer, for usps, mnist and protein, the feature dimensions are reduced to 50 by principle component analysis (PCA), then the metrics are learned in the PCA subspace. In order to avoid the influence carried by the scale of the features to the metric to be learned, the normalization procedure is applied to each feature in our experiments. In our study for mnist, we use the standard training/testing split provided by the original dataset. For the other datasets, we randomly select 70% of the data for training and the remaining 30% for testing; the experiments related to these datasets are repeated ten times, and their averaged prediction results are reported. We implement all the experiments in MATLAB on a laptop with 2.5 GHz CPU and 8 GB RAM.

Firstly, we investigate the margin distribution on the learned metrics and compare the generalization performance of nearest neighbor classification using the learned metrics and the standard Euclidean distance. In this experiment, two representative metric learning algorithms are employed: LMNN [4] and FrobMetric [30]. FrobMetric formulates the metric learning problem to use the Frobenius regularization. Actually, both of them can be treated as margin based algorithms and many empirical evidences have demonstrated they generally perform better than other metric learning algorithms for NN [4,11,30,6]. LMNN is one of the most widely-used Mahalanobis distance learning methods and has been the origin of many extensions. The training samples of LMNN are directly derived from data label: the k-nearest neighbors of any training instance should belong to the correct class while keeping away instances of other classes. FrobMetric is another popular metric learning algorithm which learns the metric with a collection of flexible triplet constraints, i.e. (x_i, x_j, x_k) where x_i is similar to x_j, and is dissimilar to x_k. Besides FrobMetric, metric learning with Boosting (BoostMetric) and information theoretic metric learning (ITML) algorithms are all of this kind [31,32,5].

In our experiment, the number of nearest neighbors k is set to 1 for LMNN. Meanwhile, the triplet constraints in FrobMetric are constructed using the 1-NN selection strategy: for any instance, we find two of its nearest neighbors measured by the standard Euclidean distance in the original feature space with the different and the same labels respectively. To maintain consistency with the

Table 2
The test errors and average margins of different metrics on each dataset.

Dataset	Test error			Average margin		
	Euclidean	LMNN	FrobMetric	Euclidean	LMNN	FrobMetric
SPECTF Heart	0.2819	0.2562	**0.2275**	0.0167	0.0413	**0.1814**
sonar	0.1742	0.1661	**0.1589**	0.0355	0.0452	**0.0997**
ILPD	0.3555	0.3543	**0.3293**	0.0115	0.0114	**0.0256**
diabetes	0.2657	**0.2626**	0.2643	0.0099	0.0111	**0.0136**
cardiotocography	0.2429	0.2323	**0.2148**	0.0209	0.0278	**0.0294**
segmentation	0.0450	**0.0388**	0.0398	0.0641	0.0888	**0.1288**
satellite	0.1150	0.1004	**0.0975**	0.0331	0.0718	**0.1054**
letter	0.0512	0.0392	**0.0384**	0.0285	0.0308	**0.0333**
dna	0.2494	0.1297	**0.1174**	0.0168	0.0240	**0.0279**
usps	0.0361	**0.0269**	0.0291	0.0281	0.0334	**0.0362**
mnist	0.0324	0.0273	**0.0265**	0.0259	0.0286	**0.0308**
protein	0.4405	0.4330	**0.3967**	0.0014	0.0018	**0.0021**

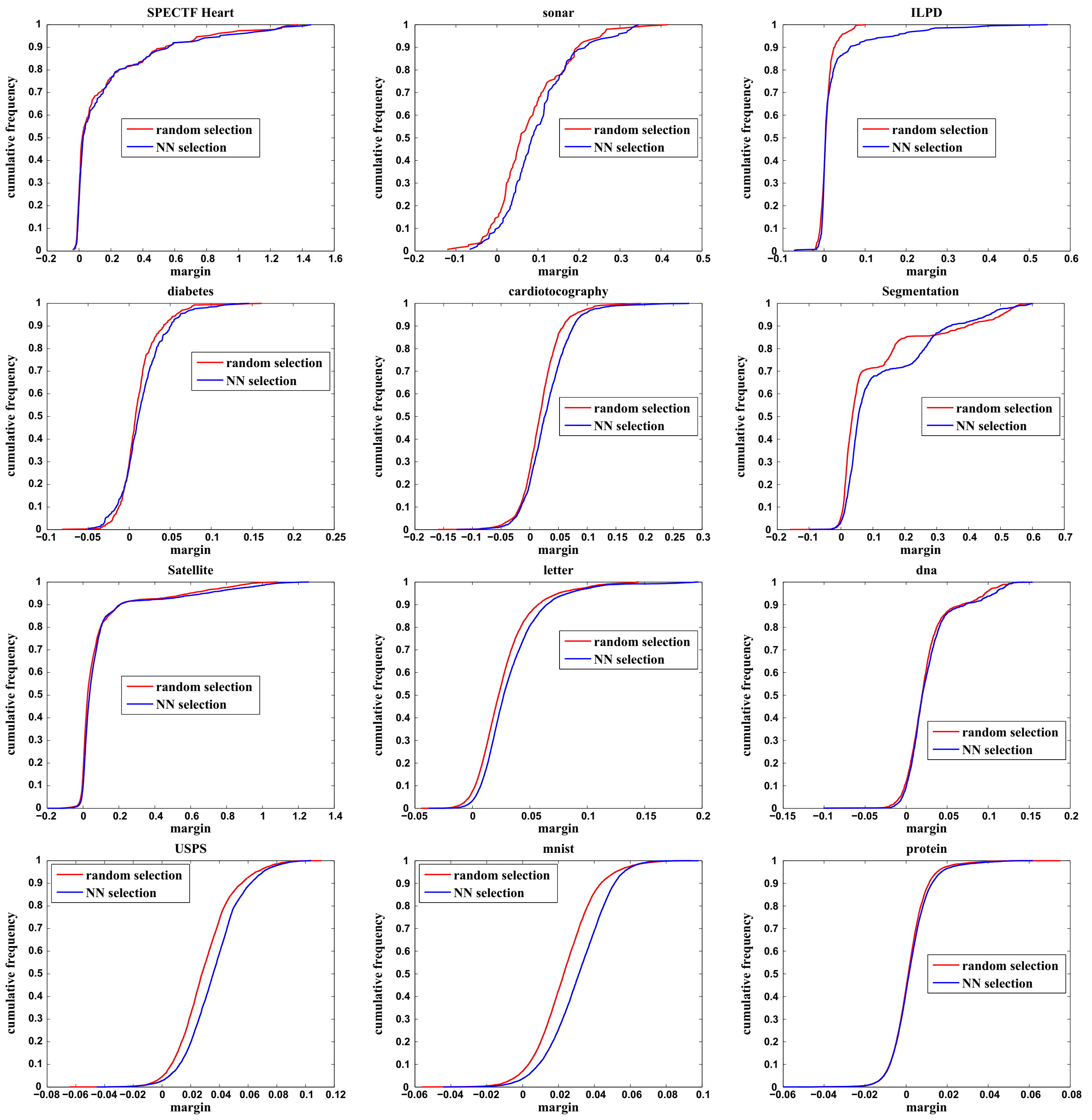


Fig. 2. Cumulative frequency (y-axis) with respect to the margin (x-axis) of random selection and NN selection strategies on all the datasets. The more right the curve, the larger the accumulated margin.

previous section, we check the generalization performance of the NN classification with the learned metrics using 1-NN. For LMNN, we use the codes provided by its authors, and also use their default parameters in our experiment. For FrobMetric, we implement the algorithm as described in [30], and the tradeoff parameters C is set to 1, unless otherwise specified. The result using the standard Euclidean distance as metric is used as a baseline.

The experimental results are shown in Fig. 1 and Table 2, respectively. Fig. 1 shows the cumulative margin distribution of Euclidean, LMNN and FrobMetric on all the datasets. Here, the margins of LMNN and FrobMetric are computed from the learned metrics such that the *nearhit* and the *nearmiss* are all measured by the learned metrics. For a fair comparison, the traces of the learned metric matrices have all been enforced to 1 to remove the scale ambiguity. It should be noted that, slightly different from the margin defined in Boosting, the range of values of our margin is not normalized. The point where a curve and the x-axis crosses corresponds to the minimum margin. As can be seen, the minimum margin of LMNN or FrobMetric is sometimes smaller than that of Euclidean distance, whereas the curves of LMNN and

Table 3
The test errors and the average margins of different strategies on each dataset.

Dataset	Test error		Average margin	
	Random selection	NN selection	Random selection	NN selection
SPECTF Heart	0.2288	**0.2225**	0.1610	**0.1728**
sonar	0.1742	**0.1629**	0.0810	**0.1003**
ILPD	0.3333	**0.3264**	0.0070	**0.0245**
diabetes	0.2757	**0.2643**	0.0117	**0.0153**
cardiotocography	0.2738	**0.2148**	0.0193	**0.0300**
segmentation	0.0646	**0.0401**	0.1059	**0.1278**
satellite	0.1404	**0.0971**	0.0918	**0.1053**
letter	0.0844	**0.0383**	0.0274	**0.0334**
dna	0.1173	**0.1169**	0.0257	**0.0281**
usps	0.0440	**0.0289**	0.0296	**0.0361**
mnist	0.0296	**0.0265**	0.0235	**0.0308**
protein	0.4081	**0.3957**	0.0015	**0.0022**

FrobMetric generally lie on the right side, showing that the margin distributions of LMNN and FrobMetric are generally better than that of the standard Euclidean distance. Table 2 shows the test errors and the average margins of different metrics on each dataset, and the best results are in bold. It is obvious that the test errors of both LMNN and FrobMetric on all the datasets are lower than that of the original Euclidean distance, and the average margins of both LMNN and FrobMetric are correspondingly larger than that of the standard Euclidean distance. The result indicates a similar conclusion as in the study of Boosting that the margin distribution rather than a single margin is really crucial for the generalization performance.

For many metric learning algorithms, the constraints can usually be constructed using different strategies, such as the NN selection or the random selection strategies. In [27], the experimental results show that the NN selection strategy is better in the construct of the constraints than the random one. To verify the conclusion obtained from their experimental results, we can investigate the margin distribution with the metrics learned from the constraints constructed with different strategies.

In this experiment, FrobMetric is employed again, as it can learn a metric from the triplet constraints flexibly constructed using different strategies. We construct the triplet constraints using the NN selection strategy and the random selection strategy respectively according to the description in [27]:

(1) *NN selection*: For any training sample (x_i, y_i), we find two of its nearest neighbors x_j and x_k measured by the standard Euclidean distance in the original feature space, with x_j sharing the same class label as x_i, and x_k belonging to a class different from x_i, and we obtain such a triplet constraint (x_i, x_j, x_k). By constructing all such triplets from the training samples, we build a triplet set using the NN strategy.
(2) *Random selection*: Given a set of training samples, we randomly select the triplets from all the possible triplets.

In our experiment, the number of constraints constructed using different strategies are both equal to the number of the training sample on each dataset.

The experimental results are shown in Fig. 2 and Table 3, respectively. Fig. 2 plots the cumulative margin distributions of the random selection and the NN selection strategies on all the datasets. As can be seen, the curves of the NN selection strategy generally lie on the right side, showing that the margin distributions of the NN selection strategy are generally better than that of the random selection strategy. Table 3 shows the test errors and the average margins of different strategies on each dataset, and the best results are also in bold. It is obvious that the overall performance of the NN selection strategy is superior to the random one, and the average margin of the NN selection strategy is larger than that of the random one. The result also indicates that although the standard Euclidean distance may not be the optimal distance function, it is still reasonable to select the initial nearest neighbors measured by the standard Euclidean distance in the absence of prior knowledge.

5. Empirical evidence with directly optimizing the margin

As can be seen in the previous section, the experimental results demonstrate that the existing algorithms LMNN and FrobMetric can obtain larger margin distributions and clearly improve the generalization performance of nearest neighbor classification on the experimental datasets. In order to further investigate the changing of the generalization performance of NN with enlarging the margin distribution, we also present a novel metric learning algorithm by directly optimizing the margin in this section. The design is inspired by the feature selection algorithm Simba [20], so we call our algorithm as SimbaMetric. It is actually a margin based metric learning algorithm for nearest neighbor classification using stochastic gradient ascent and is naturally more powerful than Simba due to the fact that it is applicable for any linear transformation of the original feature space.

In order to obtain a larger margin distribution, we wish to learn such a metric that more of the training examples have large margins. Thus, we define a margin based evaluation function according to the definition of margin of 1-NN with the Mahalanobis metric.

Definition 3. Given a set of training examples S and a positive semi-definite metric matrix M, the evaluation function is

$$e(M) = \sum_{x \in S} \rho^M_{S \backslash x}(x) = \frac{1}{2} \sum_{x \in S} (\| x - nearmiss(x) \|_M - \| x - nearhit(x) \|_M) \tag{3}$$

where $\rho^M_{S \backslash x}(x)$ is the margin of each training example x, which is calculated with respect to the training examples excluding x.

Both LMNN and FrobMetric have employed a similar evaluation function as (3). However, in the absence of prior knowledge, the real *nearhit* and *nearmiss* for each training example are unknown in advance. The choice of LMNN and FrobMetric is to use the standard Euclidean distance to determine the *nearhits* and *nearmisses*. While these *nearhits* and *nearmisses* are fixed during the learning process, the actual nearest neighbors may change as a result of the linear transformation of the input space. Although multi-pass strategy has been considered both in LMNN and BoostMetric, it is quite time consuming. To make up this weakness, we use the stochastic gradient ascent method over $e(M)$, which is a trade-off between the computational efficiency and the updating of the nearest neighbors. At each iteration, we randomly select a training example x from set S and calculate its *nearhit* and *nearmiss* with the updated metric currently. The same training example can be picked up repeatedly during the different iteration rounds. The gradient of (3) with respected to the picked training example x_i can be calculated as follows:

$$\frac{\partial e_i(M)}{\partial M} = \frac{1}{2}\left(\frac{(x_i - nearmiss(x_i))(x_i - nearmiss(x_i))^T}{\| x_i - nearmiss(x_i) \|_M} - \frac{(x_i - nearhit(x_i))(x_i - nearhit(x_i))^T}{\| x_i - nearhit(x_i) \|_M} \right) \tag{4}$$

We refer to the Mahalanobis distance matrix at the t-th iteration as M_t. At each iteration, the metric matrix can be updated by

$$M_t = Ps\left(M_{t-1} + \eta \frac{\partial e_i(M)}{\partial M} \right) \tag{5}$$

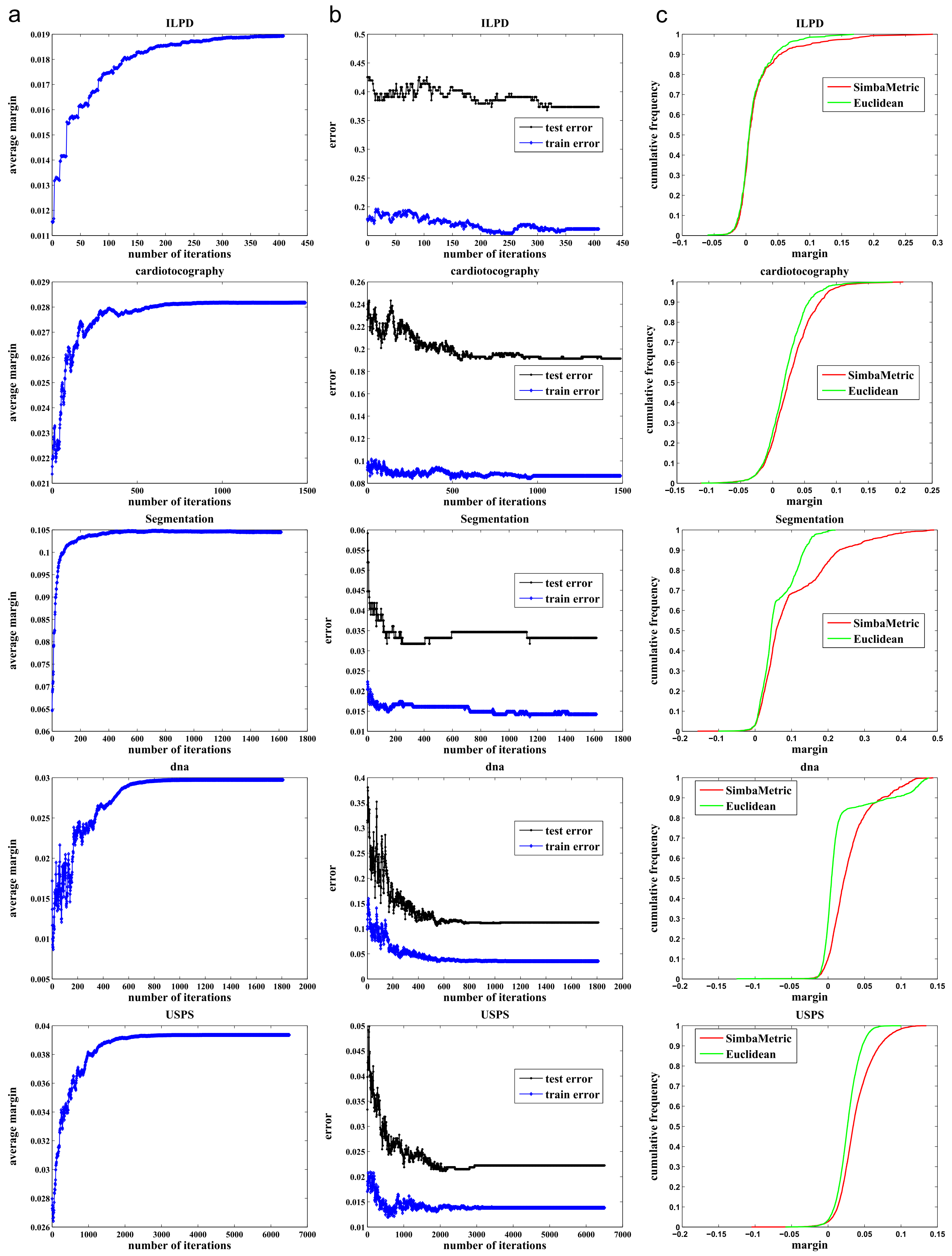


Fig. 3. The training errors and the test errors at each iteration and the average margins on different datasets.

where $Ps(\cdot)$ is the projection of the learned metric onto the cone of positive semi-definite matrices, and η is a positive step-size constant, denoting the learning rate. To remove the scale ambiguity of the learned metric, we enforce the trace of the matrix to 1 at each iteration as follows:

$$M_t \leftarrow M_t / Tr(M_t). \tag{6}$$

According to the above details, our proposed algorithm is illustrated in Algorithm 1.

Algorithm 1. Iterative Search Margin Based Metric learning (SimbaMetric).

```
Input: A set of training examples $S$, the number of iterations $T$.
1: Initialize metric matrix $M_0 \leftarrow I^{d \times d}$;
2: for $t = 1 \ldots T$ do
3:   pick randomly an training example $x_i$ from $S$;
4:   calculate $nearmiss(x_i)$ and $nearhit(x_i)$ with respect to $S$
     excluding $x_i$ and the current metric $M_{t-1}$;
5:   calculate the gradient of (3) with respected to $x_i$ according
     to (4);
6:   update the metric $M_t$ according to (5);
7:   enforce the trace of the matrix $M_t$ to 1 according to (6);
8: end for
Output: $M_t$.
```

In our algorithm, we use 20 percent of the training data as a small hold-out set to monitor the average margin, and T is set to the number of training data. In iteration of our algorithm, we initially use a relatively large learning rate of 0.5. Similar to [4], at each iteration, we increase the rate η by a factor of 1.01 if the average margin on the validation set get increased, and decrease η by a factor of 0.5 otherwise. Actually, with the increase of the number of iterations, the learning rate tends to get smaller, our algorithm using stochastic gradient ascent can typically converge. If the initial learning rate is set to too large and at the same time, no strategy is adopted to decrease it with the increase of the number of iterations, then the average margin more likely fluctuates wildly and thus makes the algorithm fail in convergence. We repeat the algorithm 5 times and obtain the learned metric with the largest average margin on the training data as the final metric.

We verify the behavior of SimbaMetric on the same datasets as in the previous section. To investigate the changing of the margin distribution and the generalization performance of NN with updating the metric, we monitor the average margin of the training set, training error and test error at each iteration on each dataset. Fig. 3 presents the results on some representative datasets, where (a) plots the average margin of the training set calculated with the updated metric at each iteration, (b) plots the training error (bottom) and test error (top) curves and (c) plots the cumulative margin distribution of SimbaMetric with the metric learned finally. The curves for the other datasets are similar. Although we only pick one training example at each iteration, the average margins on different datasets all get larger with the increasing of the number of iterations. At the same time, 1-NN trends to achieve lower error rates both on the training set and the test set with the increasing of the number of iterations. It is obvious that, across datasets, our algorithm trends to produce margin distribution graphs of roughly the same character as that of LMNN and FrobMetric in the previous section, that more of the training examples have larger margins when using the learned metric than the standard Euclidean distance.

The running time of the different metric learning algorithms is reported in Table 4. Though both of these experiments are implemented in Matlab, it is still difficult to make the comparison absolutely fair, due to the fact that the solver of LMNN is sophisticated and the main part of FrobMetric is optimized in Fortran which is more efficient than Matlab, while SimbaMetric is entirely implemented in Matlab. In spite of this, SimbaMetric is still comparable with LMNN and FrobMetric on the run time efficiency.

The test error of 1-NN with the metric learned by SimbaMetric is reported in Table 5. To compare the performance of SimbaMetric with the other methods in the previous section, statistical test on the datasets is shown in Fig. 4. According to the classification error rates listed in Tables 2 and 4, and following the statistical test setting in [33], we perform the Bonferroni–Dunn test at the significance level $p = 0.05$. The Bonferroni–Dunn test result indicates that the classification performance of SimbaMetric is statistically

Table 4
The running time on each dataset (s).

Dataset	LMNN	FrobMetric	SimbaMetric
SPECTF Heart	0.7522	0.3104	0.1888
sonar	0.5353	0.3824	0.1952
ILPD	2.054	0.1746	0.2043
diabetes	3.638	0.2257	0.1915
cardiotocography	34.95	1.255	2.102
segmentation	36.78	1.243	2.746
satellite	62.14	6.582	22.67
letter	444.8	17.01	69.54
dna	56.61	4.006	4.671
usps	223.9	23.24	62.42
mnist	17493	1541	2862
protein	360.2	95.92	274.7

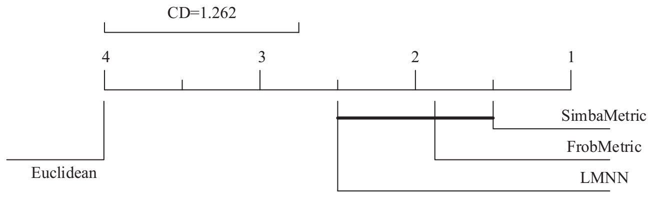


Fig. 4. Performance comparison of different metric learning methods on the Bonferroni–Dunn test. Groups of methods that are not significantly different (at $p = 0.05$) are connected. CD refers to the critical difference between the average ranks of two methods. In our experiment, the average ranks of Euclidean distance, LMNN, FrobMetric and SimbaMetric are 4, 2.50, 1.83 and 1.5 respectively.

Table 5
The test error of SimbaMetric on each dataset.

Dataset	SPECTF Heart	sonar	ILPD	diabetes	cardiotocography	segmentation
Test error	0.2362	0.1540	0.3523	0.2617	0.2203	0.0369
Dataset	satellite	letter	dna	usps	mnist	protein
Test error	0.0926	0.0392	0.1089	0.0289	0.0240	0.4188

better than that of the standard Euclidean distance at $p = 0.05$, and there is no statistically significant difference between the classification performance of SimbaMetric, LMNN and FrobMetric.

6. Conclusion

In this paper, we analyze the effectiveness of metric learning algorithms for NN from the perspective of margin distribution theoretically and experimentally. By defining the margin of 1-NN on the Mahalanobis metric, we derive a generalization error upper bound for 1-NN. The experiments on several benchmark datasets using existing metric learning algorithms demonstrate that large margin distribution can be obtained by these algorithms and it is very beneficial to reduce the generalization error. To provide more empirical evidences, a novel metric learning algorithm is also presented by directly optimizing the margin and achieves very competitive results in terms of 1-NN classification error rate, and it further verifies the effectiveness of margin on establishing the link between the learned metrics and the generalization perform ance of NN.

In conclusion, we find that the margin theory can well explain the effectiveness of metric learning algorithms for nearest neighbor classification, which is consistent with our inference. We draw a more general conclusion that the margin distribution could be a more general analysis tool able to be applicable for any linear transformation of the original feature space including Mahalanobis metric learning and other feature learning methods.

In our analysis, a single parameter i.e. the average margin is still not enough to characterize the margin distributions of the different margin based metric learning algorithms. As suggested by some recent Boosting studies [23,34], it is important to consider not only the average margin but also the margin variance. We will continue to contribute on this topic.

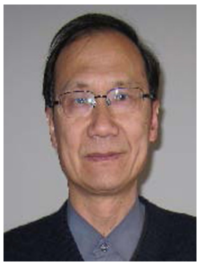
Acknowledgment

This work is partially supported by the National Science Foundation of China under Grant (61139002 and 61501229), the Fundamental Research Funds for the Central Universities under Grant NS2015091, and the Postdoctoral Funding Scheme of Jiangsu Province 1301013A.

References

[1] B. Kulis, Metric learning: a survey, Found. Trends Mach. Learn. 5 (2012) 287–364.
[2] Y. Miao, X. Tao, Y. Sun, Y. Li, J. Lu, Risk-based adaptive metric learning for nearest neighbor classification, Neurocomputing 156 (2015) 33–41.
[3] E.P. Xing, M.I. Jordan, S.J. Russell, A.Y. Ng, Distance metric learning with application to clustering with side-information, in: Advances in Neural Information Processing Systems, vol. 15, 2003, pp. 521–528.
[4] K.Q. Weinberger, L.K. Saul, Distance metric learning for large margin nearest neighbor classification, J. Mach. Learn. Res. 10 (2009) 207–244.
[5] J.V. Davis, B. Kulis, P. Jain, S. Sra, I.S. Dhillon, Information-theoretic metric learning, in: Proceedings of the 24th International Conference on Machine Learning, ICML '07, 2007, pp. 209–216.
[6] S. Trivedi, D. Mcallester, G. Shakhnarovich, Discriminative metric learning by neighborhood gerrymandering, in: Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, K. Weinberger (Eds.), Advances in Neural Information Processing Systems, 27, 2014, pp. 3392–3400.
[7] J. Lu, X. Zhou, Y.-P. Tan, Y. Shang, J. Zhou, Neighborhood repulsed metric learning for kinship verification, IEEE Trans. Pattern Anal. Mach. Intell. 36 (2014) 331–345.
[8] D. Garreau, R. Lajugie, S. Arlot, F. Bach, Metric learning for temporal sequence alignment, in: Advances in Neural Information Processing Systems, vol. 27, 2014, pp. 1817–1825.
[9] B. Mokbel, B. Paassen, F.-M. Schleif, B. Hammer, Metric learning for sequences in relational lvq, Neurocomputing 169 (2015) 306–322.
[10] A. Bellet, A. Habrard, M. Sebban, A survey on metric learning for feature vectors and structured data, CoRR abs/1306.6709, 2013.
[11] R. Jin, S. Wang, Y. Zhou, Regularized distance metric learning: theory and algorithm, in: Advances in Neural Information Processing Systems, vol. 22, 2009, pp. 862–870.
[12] A. Bellet, A. Habrard, Robustness and generalization for metric learning, Neurocomputing 151 (2015) 259–267.
[13] Q. Cao, Z. Guo, Y. Ying, Generalization bounds for metric and similarity learning, CoRR abs/1207.5437, 2012.
[14] A. Bellet, A. Habrard, M. Sebban, Similarity learning for provably accurate sparse linear classification, in: International Conference on Machine Learning (ICML-12), 2012, pp. 1871–1878.
[15] Z.-C. Guo, Y. Ying, Guaranteed classification via regularized similarity learning, Neural Comput. 26 (2014) 497–522.
[16] S. Sun, Q. Chen, Hierarchical distance metric learning for large margin nearest neighbor classification, Int. J. Pattern Recognit. Artif. Intell. 25 (2011) 1073–1087.
[17] Z.-H. Zhou, Large margin distribution learning, in: Artificial Neural Networks in Pattern Recognition, Lecture Notes in Computer Science, vol. 8774, 2014, pp. 1–11.
[18] T.M. Cover, P.E. Hart, Nearest neighbor pattern classification, IEEE Trans. Inf. Theory 13 (1967) 21–27.
[19] K. Crammer, R. Gilad-Bachrach, A. Navot, N. Tishby, Margin analysis of the lvq algorithm, in: Advances in Neural Information Processing Systems, vol. 15, 2003, pp. 479–486.
[20] R. Gilad-Bachrach, A. Navot, N. Tishby, Margin based feature selection-theory and algorithms, in: Proceedings of the Twenty-First International Conference on Machine Learning (ICML-04), 2004, pp. 43–48.
[21] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (1995) 273–297.
[22] R.E. Schapire, Y. Freund, P. Bartlett, W.S. Lee, Boosting the margin: a new explanation for the effectiveness of voting methods, Ann. Stat. 26 (1998) 1651–1686.
[23] W. Gao, Z.-H. Zhou, On the doubt about margin explanation of boosting, Artif. Intell. 203 (2013) 1–18.
[24] B. Gu, V.S. Sheng, K. Tay, W. Romano, S. Li, Incremental support vector learning for ordinal regression, IEEE Trans. Neural Netw. Learn. Syst. 26 (2015) 1403–1416.
[25] I. Guyon, S. Gunn, M. Nikravesh, L.A. Zadeh, Feature extraction: foundations and applications, vol. 207, Springer-Verlag, New York, 2008.
[26] P.L. Bartlett, The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network, IEEE Trans. Inf. Theory 44 (1998) 525–536.
[27] F. Wang, W. Zuo, L. Zhang, D. Meng, D. Zhang, A kernel classification framework for metric learning, IEEE Trans. Neural Netw. Learn. Syst. 26 (2015) 1950–1962.
[28] A. Frank, A. Asuncion, et al., Uci machine learning repository, 2010.
[29] C.-C. Chang, C.-J. Lin, Libsvm: a library for support vector machines, ACM Trans. Intell. Syst. Technol. (TIST) 2 (2011) 27.
[30] S. Chunhua, K. Junae, L. Fayao, W. Lei, A. van den Hengel, Efficient dual approach to distance metric learning, IEEE Trans. Neural Netw. Learn. Syst. 25 (2014) 394.
[31] C. Shen, J. Kim, L. Wang, A. van den Hengel, Positive semidefinite metric learning using boosting-like algorithms, J. Mach. Learn. Res. 13 (2012) 1007–1036.
[32] C. Shen, J. Kim, L. Wang, Scalable large-margin Mahalanobis distance metric learning, IEEE Trans. Neural Netw. 21 (2010) 1524–1530.
[33] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.
[34] P.K. Shivaswamy, T. Jebara, Variance penalizing adaboost, in: Advances in Neural Information Processing Systems, vol. 24, 2011, pp. 1908–1916.

Peng-Cheng Zou received the B.E. degree from the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2011. He is currently pursuing the Ph.D. degree with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing. His current research interests include machine learning and data mining.



Jiandong Wang received the Graduate degree from the Radio Department, Shanghai Jiaotong University, Shanghai, China, in 1967. He is currently a Professor and a Ph.D. Supervisor of the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China. His current research interests include machine learning, data mining, and information security.

Songcan Chen received the B.S. degree from Hangzhou University (now merged into Zhejiang University), the M.S. degree from Shanghai Jiao Tong University and the Ph.D. degree from Nanjing University of Aeronautics and Astronautics (NUAA) in 1983, 1985, and 1997, respectively. He joined in NUAA in 1986, and since 1998, he has been a full-time Professor with the Department of Computer Science and Engineering. He has authored/co-authored over 170 scientific peer-reviewed papers and ever obtained Honorable Mentions of 2006, 2007 and 2010 Best Paper Awards of Pattern Recognition Journal respectively. His current research interests include pattern recognition, machine learning, and neural computing.

Haiyan Chen received the B.S. degree, the M.S. degree and the Ph.D. degree from Nanjing University of Aeronautics and Astronautics (NUAA) in 2002, 2005 and 2012, respectively. From 2005 to 2006, she worked as a teaching assistant at College of Computer Science and Technology of NUAA and as a lecturer since 2007. In Decemeber 2012, she entered the post doctoral research station of Software Engineering in NUAA worked as a postdoctor. Her current research interests include data mining algorithm and its applications in civil aviation informatization.