NNMap: A method to construct a good embedding for nearest neighbor classification

Jing Chen [a,b,*], Yuan Yan Tang [a,b], C. L. Philip Chen [a], Bin Fang [b], Zhaowei Shang [b], Yuewei Lin [c]

[a] Faculty of Science and Technology, University of Macau, Taipa, Macau, China
[b] Chongqing University, Chongqing, China
[c] University of South Carolina, Columbia, USA

ABSTRACT

This paper aims to deal with the practical shortages of nearest neighbor classifier. We define a quantitative criterion of embedding quality assessment for nearest neighbor classification, and present a method called NNMap to construct a good embedding. Furthermore, an efficient distance is obtained in the embedded vector space, which could speed up nearest neighbor classification. The quantitative quality criterion is proposed as a local structure descriptor of sample data distribution. Embedding quality corresponds to the quality of the local structure. In the framework of NNMap, one-dimension embeddings act as weak classifiers with pseudo-losses defined on the amount of the local structure preserved by the embedding. Based on this property, the NNMap method reduces the problem of embedding construction to the classical boosting problem. An important property of NNMap is that the embedding optimization criterion is appropriate for both vector and non-vector data, and equally valid in both metric and non-metric spaces. The effectiveness of the new method is demonstrated by experiments conducted on the MNIST handwritten dataset, the CMU PIE face images dataset and the datasets from UCI machine learning repository.

© 2014 Elsevier B.V. All rights reserved.

1. Introduction

The nearest neighbor (NN) classification is a basic task of learning problems [1,2]. It classifies a query object into the category as its most nearest example. The k nearest neighbor (k-NN) classifier extends this idea by taking k nearest points and assigning the sign of the majority. Since the k-NN contains the NN as its special case with $k=1$, we do not distinguish the two terms in this paper. Since its simplicity and efficiency in practical applications, NN classifier has attracted many researchers to make efforts [3], and is applied in various domains [4–6]. However, there are still following two main problems to limit the usage of NN classifier:

1. The efficiency of NN classification heavily depends on the type of distance measure, especially in a large-scale and high-dimensional database. In some applications, such as handwritten digit recognition, DNA sequence analysis, and time series matching, the data structure is so complicated that the corresponding distance measure is computationally expensive [7,8]. NN classifier has to compare it with all the other examples in the database for each query. It may become impractical for a huge database and frequent queries.

2. Embedding learning is one of the commonly used strategies to solve the first issue. Here, a natural question is which type of embedding is good for NN classifier. There are few quantitative criterions to measure the quality of embedding. Embedding quality evaluation relies more on intuitive observation for low-dimensional (e.g., 2-D and 3-D) data visualization and classifier performance evaluation (e.g., accuracy) for pattern classification. A quantitative direct quality measure of embedding is necessary for learning an embedding to construct an efficient distance.

To overcome the above limitations of NN classification, in this work, we propose a new method called as NNMap, which speeds up NN classifier through data embedding. The proposed method makes three key contributions to the current state of the art [9–14]. The first contribution is that the proposed method obtains an efficient distance metric to take place of the original expensive distance. The NN classification is performed in the embedded vector space, rather than the original space, with the cheap distance metric obtained. This will bring several benefits. Firstly, the proposed method leads to a significant improvement for the time cost of distance computations. The main part of NN classification cost lies

on distance computation. In the paper, a resulted distance metric is vector-based and easy to be computed. In addition, the experiments show that the intrinsic discriminant structure is preserved in the embedded vector space.

The second contribution is defining a new quantitative criterion of embedding quality for NN classification. By using this criterion, one can evaluate quantitatively the embedding methods for NN classification. In the past, more attentions were focused on how to design a complicated objective function of a classifier, rather than the formal assessment of embedding quality for classification. Embedding quality evaluation relies more on intuitive observation, such as linear discriminant analysis [15], locality preserving projection [12] and graph embedding [16] do. We believe that the quantitative criterions of embedding quality would be various with respect to diverse application environments. With classification problems to be considered, it should be defined based on the classifier used. This paper proposes a formal quantitative criterion of embedding quality, which is designed specially for NN classification. It directly measures how well the embedding improves the purity degree of class label in the neighborhood of each training object. The larger local purity leads to the better performance of NN classification. So, under this definition, an embedding with good quality is expected to be good for NN classification.

The third contribution is a supervised embedding optimization method (NNMap) under the ensemble learning framework with the proposed quantitative criterion. The learning procedure follows the idea of the work of Athitsos et al. [18], which constructs a multi-dimensional embedding through the boosting framework. However, different from Athitsos's work, we give a theorem to guarantee the good quality of the proposed embedding method. Additionally, classification problems are concerned here rather than retrieval application in [18]. In the proposed method, any one-dimensional sub-embedding defines a binary classifier that predicts whether the quality condition is satisfied. Under the basic idea that the linear combination of multiple weak classifiers leads to a strong classifier in the ensemble learning procedure, here the problem of constructing multidimensional embedding is reduced to the classical boosting problem of combining many sub-embeddings into an optimal final embedding. Following the boosting framework, the combination is easy to be implemented. The key question is how to design sub-embeddings which correspond to weak classifiers.

The rest of this paper is organized as follows. In Section 2, we place our work in the context of other related studies. The basic definitions used in this work are described in Section 3. The proposed embedding method is presented in Section 4. In Section 5, the bound of embedding quality is analyzed. In Section 6, experimental results are provided to illustrate the performance of the NNMap method. Finally, conclusions are given in Section 7.

2. Related work

In order to handle the above-mentioned limitations of nearest neighbor classifier, various techniques have been developed, including traditional fast NN classifications (e.g., tree-based NN and hashing-based NN), explicit embedding learning (e.g., subspace learning and manifold learning) and implicit embedding learning (e.g., distance learning). In this section, we provide a brief review on these related techniques for NN classifier. Furthermore, our proposed method is constructed under the boosting framework. So the boosting-related algorithms are also introduced in this section. Finally, at the end of this section, we shall compare our proposed method to these related techniques in order to place our method in a clear background context.

The NN classification is initially proposed by Cover and Hart [1], which can be viewed as the extreme case of prototype-based classification with all training data to be prototypes. The NN classifier has been widely extended. For example, it has been extended for uncertain data under the belief function framework [19,20]. However, as the first limitation summarized above, the NN classifier will become sensitive and inefficient when the size of database is larger, the used distance measure is more complicated or the feature dimensionality is higher. Then, various techniques have been explored to speed up the NN classifier. To make the nearest neighbor search more efficient, various tree-based algorithms have been employed, such as VP-tree [9], KD-tree [10], BBD tree [21], R-tree variations [22,23], and Cover tree [24]. With the benefit of the tree-structured organization of data, they can reduce the number of search candidates. However, they also suffer from the high time cost and space cost for such tree construction when the dataset is large-scale and high-dimensional [25]. To achieve more efficiencies in time and space, several approximate nearest neighbor search algorithms have been proposed, which compromise the accuracy. The typical examples are hashing-based algorithms, such as locality-sensitive hashing [26], and spectral hashing [27], and coherency sensitive hashing [28]. These hashing algorithms first project data onto low-dimensional subspaces and search for approximate nearest neighbors in projected subspaces through hash tables to reduce search space significantly. However, these techniques need additional high cost to be implemented for the exact nearest neighbor search.

Another large group of techniques to expedite NN classifier especially in high-dimensional feature space, which is closely related to our method, is embedding learning. They embed objects into another space with a better structure, and can be summarized into two categories: explicit embedding and implicit embedding. The first category aims to obtain the explicit features in objective space with Euclidean or pseudo-Euclidean distance. This category includes a large amount of work, such as subspace learning and manifold learning. The other category directly learns an optimal distance measure without constructing explicit objective features, i.e., distance learning. The NNMap combines these two sides, which not only gives the explicit feature vectors in embedded space but also leads to an efficient distance metric.

Subspace learning is a class of early and most common embedding methods, which is applied in many domains such as face recognition [29]. Principal Component Analysis (PCA) seeks a set of orthogonal bases in low-dimension space to approximate original data in L2 norm sense [11]. Non-negative Matrix Factorization (NMF) follows this approximation idea with additional non-negativity constraints [30]. The NMF method is often improved by adding supervised information for pattern recognition tasks. The most famous one of subspace methods is Linear Discriminant Analysis (LDA). It builds transformation matrix to separate the inter-class scatter and compress the intra-class scatter simultaneously. The PCA, NMF, and LDA are all linear methods. To handle nonlinear structures in data space, the kernel trick is introduced. The kernel-based algorithms map original data into a linear space in higher dimensionality inexplicitly, and apply linear subspace learning methods in projected space. All linear subspace methods mentioned above can be modified into their kernel versions, such as Kernel PCA [31], Kernel NMF [32], and Kernel LDA [33]. It needs to note that kernel methods have no explicit embeddings [34]. Our method pursues explicit feature vectors, which are necessary for NN Classifier.

For dimensionality reduction and data visualization, manifold learning is proposed. Locally Linear Embedding [35], Isomap [36] and Laplacian eigenmap [37] aim to construct low-dimension data embeddings, and meanwhile preserve the intrinsic data structure locally. These methods learn objective features in a low-dimension

manifold. However, there is no explicit projection model for mapping a query object into the low-dimension manifold. It is the most limitation for the usage of these methods. To overcome this problem, Locality Preserving Projection (LPP) constructs an explicit linear transformation following the manifold embedding idea. Recently, Yan et al. summarized most of previous dimensionality reduction algorithms to propose a general uniform framework, named Graph Embedding (GE) [16,17]. Although the NNMap has no transformation matrix, it is easy to compute the embedding vector for a query object.

Distance learning starts from the work of Xing et al. The NNMap also results in an efficient distance metric when multi-dimension feature vectors are obtained. Many earlier distance learning algorithms, like Xing's method [13], Relevant Component Analysis (RCA) [38], Discriminative Component Analysis (DCA) [39], and their kernel versions, learn general Euclidean distance with global constraints. Recently, many methods focus on the local structure, such as Discriminant Adaptive Nearest Neighbor [40], Neighborhood Components Analysis (NCA) [14], Large Margin Nearest Neighbor (LMNN) [41], and many other studies [42,43]. The NNMap is designed for NN classification, and it focuses on the local structure. Furthermore, it requires no additional side constraints except for original sample data and corresponding labels.

The final related techniques to our proposed method are Boosting-based algorithms. Boosting [44] and Bagging [45] are the two of the commonly used techniques for ensemble learning. Compared with Bagging, Boosting performs better in most cases [46]. AdaBoost [47] is one of the most popular Boosting-based algorithms, which has been widely applied to many pattern recognition fields, such as face recognition. AdaBoost creates a collection of weak learners, and corrects the misclassifications of weak classifiers alternatively by adjusting their corresponding weights adaptively: increase the weights of the samples which are misclassified by current weak learner and decrease the weights of the samples which are correctly classified. Many other classifiers, such as nearest neighbor classifier, decision trees [48], and neural networks [49], have been used as weak learners for AdaBoost. Originally, AdaBoost was developed for binary classification problems. Later, it was extended to multi-class cases, which are called as AdaBoosl.M1 and AdaBoost.M2 [47]. Except for classification applications, some variations of AdaBoost have been proposed for regression problems by projecting regression problems into classification problems, for example, AdaBoost.R [47] and AdaBoost.R2 [50]. In this work, we extend the application of AdaBoost to embedding learning.

Neither subspace learning nor manifold learning pays little attention on defining a quantitative criterion for embedding quality. For example, LDA characterizes the intra-class compactness and inter-class separability, simultaneously. However, its embedding quality is just measured in the intuitive sense, rather than by a formal criterion. In this paper, we define a quantitative criterion. The embedding method is designed based on this quantitative criterion. Most of previous studies for embedding quality, in particular, Lipschitz embeddings [51], FastMap [52], MetricMap [53], and SparseMap [54], address embedding optimization from a global geometric perspective. In contrast, our method constructs optimal embedding preserving local discriminant relations. For exploring the local structures of the data, there are many works, for example, [55,56]. However, those presented local structures are not designed for embedding quality evaluation.

Finally, the need for being able to handle novel data types is becoming ever more important in modern pattern recognition application. Non-vector and non-metric problem require more attention. Most of methods above mentioned can only address vector or tensor data. Typical subspace learning methods such as PCA, LDA and NMF, manifold learning algorithms such as LLE, LPP, and GE, distance learning approaches such as RCA, DCA, NCA and LMNN, embedding optimization such as FastMap, MetricMap and SparseMap are all based on Euclidean distance metric, and appropriate only for vector or tensor data. Our proposed method can be applied in both metric and non-metric space, and suitable both for vector and non-vector data.

3. Background and notation

We first introduce two basic local descriptors of data distribution structure, the positive ratio and the negative ratio. Later, we give a formal definition of good embeddings. This definition is also a reasonable measure for the discriminative capability of the embedded data distribution structure. Particularly, a good one-dimension embedding is defined as a special case of Definition 1. Boosting two types of famous one-dimension embeddings — reference object embedding and line project embedding — the NNMap method constructs a final multi-dimension embedding. Finally, we note that the quality of the final embedding is consistent with the definition of a good embedding, which is guaranteed by Theorem 1.

3.1. Embedding quality

We are given access to a database U of labeled examples $(x, l(x))$ drawn from some distribution P over $X \times \{1, 2, …, m\}$, where X is an instance space, $l(x)$ denotes the class label of x, and $U = \{(x_1, l(x_1)), (x_2, l(x_2)), …, (x_N, l(x_N))\}$. f is an embedding $f : X \to E$, where E is the embedded space. In this work, E is only considered as a real number vector space. Here, f is called as 1-dimension (1-D) embedding if $E = R^1$, and called as d-dimension (d-D) embedding if $E = R^d$. We consider the d-D embedding F formulated as $[F_1, F_2, …, F_d]$, where F_i is a 1-D embedding. An object $x \in U$ is projected as $e_i \in R^1$ by $F_i(x), i = 1, …, d$. So if d-D embedding F is applied on x, we have

$$F(x) = [F_1(x), F_2(x), …, F_d(x)] = [e_1, e_2, …, e_d]. \tag{1}$$

Definition 1. For $E = R^d$, given an example $e \in E$, the positive ratio $r^+(e)$ of x is defined as

$$r^+(e) = \begin{cases} \frac{1}{k} |\{e' | l(e) = l(e'), e' \in \mathrm{NN}(e, E, k)\}| & \text{when } e \in R \\ \sum_{i=1}^{d} \alpha_i r^+(e_i), \text{ where } e = [e_1, e_2, …, e_d] & \text{when } e \in R^d \end{cases} \tag{2}$$

Meanwhile, its negative ratio is

$$r^-(e) = \begin{cases} \frac{1}{k} |\{e' | l(e) \neq l(e'), e' \in \mathrm{NN}(e, E, k)\}| & \text{when } e \in R \\ \sum_{i=1}^{d} \alpha_i r^-(e_i), \text{ where } e = [e_1, e_2, …, e_d] & \text{when } e \in R^d \end{cases} \tag{3}$$

where $|\{\cdot\}|$ is the number of elements in the set $\{\cdot\}$, $\mathrm{NN}(e, E, k)$ denotes the set of k nearest neighbors of the example $e \in E$, and α_i's are corresponding weight coefficients, which would be determined though the embedding construction procedure later.

These two concepts are actually the descriptors of the local structure of the sample distribution. $r^+(e)$ measures the purity of class labels in a neighborhood around the example e. The more homogeneous label distribution in the neighborhood leads to the larger value of $r^+(e)$. In contrast, the less purity of labels leads to the larger value of $r^-(e)$. Based on these two local structure descriptors, we introduce the criterion of good embeddings as Definition 2. Then naturally, we can specialize the above definition into the 1-D and d-D cases.

Definition 2. For any embedding $F : X \to E$, it is said to be an (ϵ, γ)−good embedding for NN classifier, if at least $1 − \epsilon$ probability

mass of training examples x satisfy,

$$Pr[r^+(e) - r^-(e) > 0] \geq 1 - \gamma, \tag{4}$$

where $e = F(x)$, $0 \leq \gamma \leq 1$.

In particular, we have that if $E = R$, the embedding F satisfied Definition 2 could be called as an (ϵ, γ)−good 1-D embedding. And F is (ϵ, γ)−good d-D embedding if $E = R^d$. This type of the definition of good embedding quality is much intuitive and direct. This definition expresses that the examples with the same class label stand closer than those from the different classes. It is also the basic assumption for NN classifier. More precisely, we use ϵ as a pseudo-loss parameter and γ as a margin between the values of r^+ and r^- to characterize quality goodness of a embedding.

3.2. Some 1-D embeddings

Given any space X, the distance function D between elements of X can be extended to the measure between elements of X and subsets of X. Let $x \in X$ and $Q \subset X$. Then,

$$D(x, Q) = \min_{q \in Q} D(x, q). \tag{5}$$

Given a subset Q of X, a 1-D Lipschitz embedding with respect to Q, $F^Q : X \rightarrow R$ can be defined as

$$F^Q(x) = D(x, Q). \tag{6}$$

It is also called a reference object embedding. Vleugels et al. suggested that Q can consist of a single object q, which is termed as a *reference object* or a *vantage object* [57]. Then the 1-D Lipschitz embedding with respect to q, $F^q : X \rightarrow R$, can be defined as follows:

$$F^q(x) = D(x, q) \tag{7}$$

Several such 1-D embeddings are concatenated as a multidimensional Lipschitz embedding.

Faloutsos [52] proposed another simple 1-D embedding, called line projection embedding. Two objects x_1, x_2 are selected, referred to as '*pivot objects*', and consider the line $\overline{x_1 x_2}$ that passes through them in original space. The 1-D embedding $F^{x_1, x_2}(x)$ can be defined by projecting elements of X onto the line as follows:

$$F^{x_1, x_2}(x) = \frac{D(x, x_1)^2 + D(x_1, x_2)^2 - D(x, x_2)^2}{2D(x_1, x_2)} \tag{8}$$

For the detailed derivation and intuitive geometric interpretation of this equation, the reader can refer to the literature [52], where line projection embeddings are used to construct FastMap embeddings.

In this work, the above two types of 1-D embeddings are used to construct the final multi-dimensional embedding. In the embedding learning procedure, vantage objects and pivot objects are selected from a subset of X, which called a reference set. For the d1-D embeddings $F_1, \ldots, F_d$ to construct the d-D embedding F, a half of them are reference object embeddings, and others are line projection embeddings.

3.3. Overview of basic idea

The basic idea of our method is to boost 1-D weak embeddings into a multi-dimension strong embedding. A d-D embedding F is good for projecting original data space into an embedded feature space with a computationally efficient distance measure, when F holds the inequality (4) as possible. To construct a good d-D embedding F, d respective good 1-D embeddings are pursued. A good 1-D embedding preserves local within-class compression and between-class scattering, and holds the inequality (4) on one dimension as possible. The 1-D embeddings are learned and integrated in the Boosting learning framework. Additionally, a weighted Manhattan (L1) distance metric is defined in the embedded feature space with the integrated weights, which is used as the distance for NN classifier.

4. Embedding learning with NNMap

For NN classifier, the class label of a query object is dependent on the label distribution of its k nearest neighbors. We can define the positive ratio r^+ and the negative ratio r^- as Definition 1. In practical application, the parameter k in k-NN classifier is usually an odd number. So, the relationships of r^+ and r^- can be summarized into two cases: $r^+ > r^-$ or $r^+ < r^-$. Intuitively, we can see that the NN classifier would classify the query object into the correct class, if $r^+ > r^-$. Otherwise, an error occurs.

The boosting framework [47] is an important type of ensemble learning frameworks. Its basic idea is to boost multiple weak classifiers to obtain a final strong classifier. In this work, a multi-dimension embedding ensemble learning scheme is proposed by following the boosting framework. Here, a 1-D embedding can be viewed as a weak classifier. The boosting procedure considers the two respects: the distribution of sample data and the diverse importance of different classifiers. In this work, we also consider these two respects in the learning procedure.

4.1. 1-D embeddings as weak classifiers

Considering a d-D embedding $F : (X, D) \rightarrow (R^d, \Delta)$, define F as

$$F(X) = [F_1(X), \ldots, F_d(X)]^T \tag{9}$$

As shown above, d1-D embeddings are ensembled to constitute a d-D embedding. To construct a good d-D embedding, it is required to find d good 1-D embeddings. If we assign the equal importance to different 1-D embeddings, there is no flexibility to fit the applicable sample data, and it is difficult to achieve a high performance. So, we define the measure Δ in the embedded space as follows:

$$\Delta(F(X_1), F(X_2)) = \sum_{i=1}^{d} \alpha_i \Delta_i(F_i(X_1), F_i(X_2)), \tag{10}$$

where Δ_i is the individual distance that corresponds to embedding F_i and α_i is the assigned weight associate with embedding F_i.

At this point, every sample can be projected as a real number by an individual 1-D embedding F_i. Then the original sample set is mapped into a set of one-dimension feature data. For every object, we can compute its positive ratio r^+ and negative ratio r^-. According to Definition 2, it needs a higher probability of $r^+(e) - r^-(e) > 0$ in 1-D embedded data space to get a better 1-D embedding quality, So we could view 1-D embedding as a weak classifier with ϵ as its pseudo-loss. The higher the accuracy is, the better the classifier is. Accordingly here, the higher the probability is, the better the embedding is. For the object e, $r^+(e)$ is larger or smaller than $r^-(e)$. Intuitively, points with the same class label are closer with each other than ones with the different class labels. So we can expect the 1-D embedding with the probability $Pr[r^+(e) - r^-(e) > 0]$ to behave as a weak classifier, and be better than random guess. In the later section, we will describe how to ensemble these 1-D embeddings (weak classifiers) into a d-D embedding (strong classifier).

4.2. Construct a good embedding using boosting

Our goal is to construct a good embedding for nearest neighbor classification, i.e., $F : (X, D) \rightarrow (R^d, \Delta)$. A reference set $C \subset U$ of candidate references and pivot objects has been prepared. Accordingly, we obtain a large pool of candidate 1-D embeddings. As

analysis above, every such 1-D embedding is involved a part of information on the structure of the original space. We expect such 1-D embeddings to behave as weak classifiers. And its pseudo-loss is defined as

$$\eta = \Sigma_{i=1}^{N} p_i \cdot \frac{1}{N} \cdot [r^+(x_i) - r^-(x_i) > 0]^+ \qquad (11)$$

where $[z]^+ = 1$ when z is true, otherwise $[z]^+ = 0$. Here, the pseudo-loss η is a computational implementation of $1 - \epsilon$ on a finite sample set. And p_i is a corresponding weight of each sample point x_i. Then our problem of good embedding constructing is reduced to the problem of learning 1-D embeddings as weak classifiers and optimizing a weighted linear combination of weak classifiers. We adopt the ensemble learning framework similar to AdaBoost to design our learning procedure [47]. The overall sequence of the proposed scheme is summarized as follows:

Algorithm — NNMap

Input: the sequence of N labeled samples $(x_1, y_1), \ldots, (x_N, y_N)$, $y_i \in Y = \{1, 2, \ldots, m\}$; the set $C \subset U$ of candidate reference objects C and $|C| = \beta$; the number of iterations d.

Initialize the weighting vector $w_i^1 = \frac{1}{N}$, for $i = 1, \ldots, N$.

for the round $t = 1, 2, \ldots, d$ **do**
1. Set
$$p^t = \frac{w^t}{\sum_{i=1}^{N} w_i^t} \qquad (12)$$

2. For each candidate 1-D embedding F_s, $s = 1, \ldots, \beta$, set $\eta_s^t = \sum_{i=1}^{N} p_i^t \cdot \frac{1}{N} \cdot [r^+(x_i) - r^-(x_i) > 0]^+$, where $[z]^+ = 1$ when z is true, otherwise $[z]^+ = 0$.

3. *WeakLearn*: search an optimal projecting direction to construct a 1-D weak good embedding in terms of the object function as

$$\eta^t \leftarrow \arg \max_{s=1,\ldots,\beta} \eta_s^t \qquad (13)$$

If $\eta^t < 1/2$, then set $d = t - 1$ and abort loop.

4. Set
$$\alpha_t = \log \frac{\eta^t}{1 - \eta^t} \qquad (14)$$

5. Update the weight vector by

$$w_i^{t+1} = w_i^t \cdot \alpha_i^{[r^+(x_i) - r^-(x_i) > 0]^+} \qquad (15)$$

end

Output: the good d-D embedding and the distance metric:

$$F(x) = [F_1(x), F_2(x), \ldots, F_d(x)]^T \qquad (16)$$
$$\Delta(x, z) = \sum_{t=1}^{d} \alpha_t |F_t(x) - F_t(z)| \qquad (17)$$

Here, let us further explain the meanings of some parameters. η_s^t is the specific case of η defined in Eq. (11) and stands for the pseudo-loss of the s-th 1-D embedding F_s which acts as a weak classifier in the t-th WeakLearn training cycle. p_i^t is the specific case of p_i, which is a weight of the sample x_i, in the t-th WeakLearn training cycle. In this work, we adopt the two types of 1-D embeddings, i.e., 1-D reference object embedding and 1-D line projection embedding, which have been explained in Section 3.2. Therefore, the set of β candidate 1-D embeddings includes $\beta/2$ 1-D reference object embeddings and $\beta/2$ 1-D line projection embeddings. To construct 1-D reference object embeddings, we first randomly select $\beta/2$ objects from the candidate reference samples C and construct 1-D reference object embeddings

in terms of Eq. (7). Then we randomly select $\beta/2$ object pairs from C to construct $\beta/2$ 1-D line projection embeddings following Eq. (8). Finally, $\beta/2$ 1-D reference object embeddings and $\beta/2$ 1-D line projection embeddings constitute the candidate 1-D embedding set.

As shown in this overall sequence, NNMap involves two main steps: training weak learners and weights update. There are three significant points that need to be emphasized.

1. The input data could be in the form of numeric vector or other complex structures, such as sequence strings, graph data, and time series. No matter how complicated the data space is, it can be input into NNMap only if the distance measure of the original data space is defined.
2. In the first step, NNMap substitutes weak 1-D embeddings learning for weak classifiers learning. In AdaBoost, weak classifiers are usually learned through adjusting parameters in classifiers. Here, a large set of candidate 1-D embeddings is ready for being selected. The training procedure is to search for the appropriate projecting direction to constitute the 1-D embedding with positive ratio r^+ as high as possible. The embedding searching process will be explained later in detail.
3. This learning algorithm is suitable for both the binary classification and the multi-class case, although AdaBoost is designed only for the binary problem. When a multi-class problem is considered, NNMap behaves like AdaBoost.M1 which is the multi-class extension of AdaBoost [47]. In NNMap embedding framework, we integrate the two cases in the definitions of r^+ and r^-, which are not particular for the binary case or the multi-class case. In Boosting learning procedure, the requirement of $\eta > \frac{1}{2}$ leads that the quality of a single 1-D embedding in the multi-class problem is required to be significantly stronger than that in the binary case. It brings unavoidable difficulties as the case of AdaBoost.M1. Here, we solve this problem through using the random-based strategy for the reference set generation. It provides candidate reference and pivot objects as many as possible.

Next, we will specify how to prepare a reference set. The common idea is to select a subset of sample data like FastMap, Lipschtz embedding and many other embedding methods do. However, this strategy is limited by the size of sample data. Particularly, in some application fields, the sample size is small, and furthermore, new samples are not easy to be generated. Therefore, it projects original data onto the very limited directions, which could not provide sufficient flexibility for embedding learning. In this work, we construct a random reference set. All reference and pivot objects are generated randomly in the range of sample data scattered. The significant advantage of this strategy is that the size of candidate reference set is unlimited. The reference and pivot objects would be generated with an arbitrary quantity. It would provide an arbitrary quantity of projecting directions to weak learning procedure to search for optimal 1-D embeddings.

Another point needed to be explained is the number k' of nearest neighbors for computing positive ratio r^+ and negative ratio r^-. We suggest that the number k' is set to be a little larger than the number k used for the final k-nearest neighbor classifier. If the distribution of class labels is homogeneous in a larger neighbor range, it tends to be also homogeneous in a smaller neighbor range. The contrary may be not true. In this work, we set $k = 1$ and $k' = 5$.

The direct output of the NNMap learning procedure is a series of 1-D good embeddings F_t and corresponding weights α_t, $t = 1, \ldots, d$, while the output of AdaBoost is a strong classifier. Our final goal is to construct a multi-dimension embedding. To achieve that, we

define an embedding $F(x) : X \to R^d$ and a distance metric Δ:

$$F(x) = [f_1(x), f_2(x), ..., f_d(x)]^T \tag{18}$$

$$\Delta(x, z) = \sum_{t=1}^{d} \alpha_t |f_t(x) - f_t(z)| \tag{19}$$

The embedding dimensionality d is the same as the iteration number T in the boosting learning procedure. Δ is a weighted Manhattan (L1) distance measure. Δ is a metric, because the training procedure guarantees that all α's are non-negative. Moreover, the operators involved in computing a vector-based distance measure are easy to be implemented. Δ is efficient on computation.

5. Theoretical analysis of NNMap embeddings

In this section, we provide theoretical analysis of NNMap embeddings. The d-D embedding obtained from the NNMap procedure is proved to be a good embedding in terms of Definition 2. Then we give the up bound of the error probability of the NNMap d-D embedding, which is equivalent to find the low bound of the NNMap embedding quality.

5.1. Theoretical guarantee for multi-dimension embedding quality

Given d (ϵ, γ)–good 1-D embeddings, as shown in the following theorem, a sufficiently good d-D embedding can be learned.

Theorem 1. *Given a set of 1-D embeddings $F_i, i = 1, ..., d$, where for any $x \in X$ we have $F_i(x) = e_i$, a direct d-D embedding in the form as*

$$F(x) = [F_1(x), F_2(x), ..., F_d(x)] = [e_1, e_2, ..., e_d] = e \tag{20}$$

can be constructed. If $F_1, F_2, ..., F_d$ are all (ϵ, γ)–good 1-D embeddings, then $d = (4/(1-2\gamma)^2)\ln(1/\delta)$ is sufficient so that with probability of at least $1 - \delta$ the following condition is satisfied:

$$Pr[r^+(e) - r^-(e) > 0] \geq 1 - (\epsilon + \delta) \tag{21}$$

Proof. The proof uses a technique in reference [58]. If $F_1, F_2, ..., F_d$ are all (ϵ, γ)–good 1-D embeddings, then for any $x \in X$, by $F_i(x) = e_i$, $i = 1, ..., d$, with $1 - \epsilon$ probability it satisfies the inequality condition (4). Let M be the set of training examples satisfying the inequality (4). So, by assumption, $Pr[x \in M] \geq 1 - \epsilon$. Now fix any $x \in M$,

$$Pr[r^+(e_i) - r^-(e_i) > 0] = E[I(r^+(e_i) - r^-(e_i) > 0)]$$
$$= \frac{1}{2} \cdot E[\text{sgn}(r^+(e_i) - r^-(e_i))]$$
$$+ \frac{1}{2} \tag{22}$$

where I denotes the indicator function. Using the above equation, inequality (4) is equivalent to

$$E[\text{sgn}(r^+(e_i) - r^-(e_i))] \geq 1 - 2\gamma \tag{23}$$

Since $0 < r^+(e_i) < 1$ and $0 < r^-(e_i) < 1$, by Chernoff bounds over d random (ϵ, γ)–good 1-D embeddings noted by C, we have that

$$Pr_C(r^+(e) - r^-(e) \leq 0)$$
$$= Pr_C\left(\frac{1}{c}\sum_{i=1}^{d}\text{sgn}[\alpha_i(r^+(e_i) - r^-(e_i))] \leq 0\right)$$
$$\leq e^{-d(1-2\gamma)^2/2} \tag{24}$$

Since the above inequality is true for any $x \in M$, we can take expectation over all $x \in M$, which implies that

$$E_{x \in M}[Pr_C(r^+(e) - r^-(e) \leq 0)] \leq e^{-d(1-2\gamma)^2/2} \tag{25}$$

Thus we have

$$E_C[Pr_{x \in M}(r^+(e) - r^-(e) \leq 0)] \leq e^{-d(1-2\gamma)^2/2} \tag{26}$$

Here, we set $\delta^2 = e^{-d(1-2\gamma)^2/2}$. Using Markov's inequality we obtain

$$Pr_C[Pr_{x \in M}(r^+(e) - r^-(e) \leq 0) \geq \delta] \leq \delta \tag{27}$$

Adding in the ϵ probability mass of points not in C yields the theorem. □

The main goal of this work is to construct an embedding for speed up NN classification and design a strategy to quantitatively measure the quality of this embedding. Theorem 1 indicates that the d-D embedding constructed by NNMap is quantitatively indeed a good embedding. From the above quality criterion, we could conclude that the embedding quality corresponds to the amount of the preferable local structure preserved by the embedding. The larger the amount of the local structure optimized, the better the embedding quality is. So the quality of local structure is used as a direct measure of embedding optimization in our proposed method.

5.2. Bound analysis for multi-dimension embedding quality

As the definition of the embedding quality, the index ϵ is a measure for embedding quality. The smaller the ϵ is, the better the embedding quality is. So ϵ is respected to be as small as possible for constructing good embedding. Here, we give the up bound of the error probability ϵ.

Theorem 2. *Suppose the WeakLearn algorithm generates 1-D embedding $F_1, F_2, ..., F_d$ with pseudo-losses $\epsilon_1, \epsilon_2, ..., \epsilon_d$. Then after NNMap is applied, the final d-D embedding $F : (X, D) \to (R^d, \Delta)$ is (ϵ, γ)–good, and ϵ is bounded by*

$$\epsilon \leq 2^d \prod_{i=1}^{d} \sqrt{\epsilon_i(1 - \epsilon_i)} \tag{28}$$

As analysis above, the indexes $\epsilon_1, ..., \epsilon_d$ and ϵ correspond to the errors of weak hypotheses and the final hypothesis of AdaBoost, respectively. So the proof of Theorem 2 can be achieved simply along the way of error bound analysis of AdaBoost. This theorem ensures that a good embedding would be constructed following our proposed NNMap embedding method, if 1-D embeddings selected are sufficiently good. It is noted that the final d-D embedding quality depends on all the respective 1-D embedding qualities. And this quality bound achieves the maximum when the qualities of all the 1-D embeddings are uniform.

6. Experiments

We applied our proposed embedding approach NNMap for classification and compared it with several alternative methods. The experiments are performed on the 10 public datasets: the MNIST for handwritten digit recognition, the CMU PIE for face recognition, and 8 UCI machine learning datasets. In the handwritten digit recognition application of MNIST database, the state-of-the-art distance measure is shape context matching [59], which measures the similarity between shapes rather than pixel brightness values. It first solves the correspondence problem between points on the two shapes, then uses the correspondences to estimate an aligning transform, and finally compute the distance between the two shapes as the sum of matching errors between corresponding points, together with a term measuring the magnitude of the aligning transformation. In the pattern recognition applications of CMU PIE database and UCI databases, for these vector features, as reported in many existing publications, the Euclidean distance is commonly used and evaluated to be efficient. So, in our experiment, we use shape context matching as the original distance for MNIST database, Euclidean distance for CMU

PIE and UCI databases. For our proposed NNMap, it transform the problem of original distance into the NN classification with a weighted Manhattan distance.

6.1. Data sets

6.1.1. Handwritten digit dataset

The MNIST dataset of handwritten digits contains 70,000 digit images for the 10 numbers from 0 to 9. Each image is cropped and resized to be 28×28 pixels, with an isolated digit in the center. A sample image is illustrated in Fig. 1. All methods are evaluated by using the five-run of 10-fold cross validation. For each run, the dataset is randomly split into 10 equal partitions. The partition process ensures each part contains 10 class labels. Nine of 10 partitions are used for learning objective embedding, and the rest one partition is used for testing. This procedure is iterated until each partition has been used as a testing set. The averaged results of the 50 executions are reported. The distance measure used for brute force method and leaning procedures in embedding methods is shape context matching. The shape context features are extracted following reference [59].

6.1.2. Human face dataset

The CMU PIE dataset contains 41,368 images of 68 people, each person under 13 different poses, 43 different illumination conditions, and 4 different expressions. We use a subset, which only contains five near frontal poses (C05, C07, C09, C27, C29) and all the images under different illuminations and expressions. So, there are 170 images for each individual except for the 38th individual with 160 images, and 11,550 images in all. The images are cropped and resized to be 32×32 pixels. A sample image is illustrated in Fig. 2. Each image is unfolded directly into a 1024-dimension vector. The five-run of 10-fold cross-validation is performed on this dataset. The distance measure used is Euclidean distance which is commonly used for vector pair comparisons.

6.1.3. UCI machine learning dataset

The eight datasets are selected from the UCI Machine Learning Repository. They are Vehicle, Spambase, Satimage, Adult, Census-income, Covertype, Poker Hand and KDD Cup 99. A description of these datasets including the number of instances (#Instances) and classes (#Classes) and the feature space dimensionality (#Att) is presented in Table 1. The set of databases covers a wide range of sizes, which ensures the reliability of experimental conclusions. The missing values in these datasets are filled by randomly selected values from the samples belonging to the same class. To reduce the possible bias of the results, 10-fold cross-validation is also performed with five-run random partition. These datasets are

Fig. 1. Samples from the MNIST database.

Fig. 2. Samples from the CMU PIE database.

Table 1
Description of UCI datasets.

Dataset	#Instances	#Att	#Classes
Vehicle	846	18	4
Spambase	4597	57	2
Satimage	6435	36	7
Adult	48,842	14	2
Census-income	299,285	39	2
Covertype	581,012	54	7
PokerHand	1,025,010	10	10
KDDCup99	4,898,431	41	23

constituted by vector objects, so Euclidean distance is adopted as the distance measure.

6.2. Comparison methods

We compare NNMap with the following alternative methods for nearest neighbor classification:

1. *Brute-force*: Firstly, a reference set C with d objects selected randomly from the whole database is constructed. For each query object, it is compared with all reference objects by using the original distance measure (shape context matching for handwritten digits and Euclidean distance for face images), and then it is classified into the class of its nearest neighbor.
2. *FastMap* [52]: We construct a d-D embedding by performing FastMap with a reference set R. The embedding dimensionality d is varied. The reference set C used for each dataset is its corresponding training set.
3. *Random object projection* (ROP): We first construct a 1-D Lipschitz embedding by choosing a random reference object p from the training set. Here, the whole training set is used as the reference set. Multiple 1-D random Lipschitz embeddings were concatenated into a multidimensional embedding. After projected vectors are obtained, Euclidean distance is used to compare all possible pairs for nearest neighbor classification.
4. *Random line projection* (RLP): We first construct a 1-D embedding by projecting a considered object onto a line, where pivot objects are chosen randomly from the training set. Multiple 1-D embeddings were concatenated into a multi-dimension embedding. And then nearest neighbor classification is performed with Euclidean distance.
5. *VP-trees* [9]: VP-trees are built to speed up Nearest Neighbor classification for large size database. However, the application of VP-trees requires that the sample space has a valid metric, which specially satisfies the triangle inequality. The distance in the MNIST experiment is nonmetric (shape context matching), and particularly, does not satisfy the triangle inequality. We adopt the modified version of VP-trees as that in [60] for MNIST. In the modified VP-trees, the triangle inequality is satisfied up to a constant ζ. To keep a good balance between the accuracy and the efficiency, we set $\zeta = 3.5$ in our experiments.
6. *Locality sensitive hashing* (LSH) [26]: It relies on a family of locality sensitive hashing functions by which sample points are mapped into hash tables. The number of hashing functions is denoted as d. For each function, the probability of collision is much higher for samples which are close to each other than for those which are far apart. Then, near neighbors could be found by hashing the query object and retrieving elements stored in buckets containing that object. There is the same problem as that of VP-trees. For MNIST, The searching algorithm is modified so that it guarantees correct neighbors if the triangle inequality is satisfied up to a constant ζ [61]. Here, $\zeta = 5$ for a good balance.

Furthermore, the typical 3-NN and 5-NN classifiers, and the typical Adaboost algorithms with NN classifiers as weak learners (abbreviated as BoostNN) are also compared with our methods:

1. *3-NN*: To enhance the comparison to brute-force methods, except for the typical 1-NN classifier (denoted as Brute-force), the performances of 3-NN and 5-NN are also compared. As the construction of 1-NN, it is performed on the reference set C of d objects randomly selected from the whole database. For each query object, it is compared with all reference objects by using the original distance, and then it is classified by major voting of its most three nearest neighbors.
2. *5-NN*: It is also performed on the reference set C of d objects randomly selected from the whole database. And it also uses the original distance measure and each query object is classified by major voting of its most five nearest neighbors.
3. *BoostNN*: It is constructed in terms of AdaBoost.M1 with nearest neighbor classifier as its weak learner [47]. Each weak learner is performed on a subset randomly selected from the training set with a 20% size. Finally, d NN weak classifiers result in a final classifier which output a final decision.

6.3. Results

We performed nearest neighbor classification in the embedded feature space constructed by the methods — NNMap, Brute-force, FastMap, ROP, RLP, VP-tree and LSH with various embedding parameters. In addition, the t-test is performed to evaluate the statistical significance of the results. Two methods with the same letter in common exhibit no statistically significant difference with 95% confidence. The corresponding result of NNMap is shown in bold.

6.3.1. Experiments on the MNIST dataset

The experimental results on MNIST dataset are summarized in Table 2. Zhang et al. use the discriminative classifier based on one prototype with shape context distances as a feature vector. From multiple prototypes, a final classifier is obtained by combining single classifiers. Zhang's method is also implemented for comparison goal. For Brute-force, 3-NN and 5-NN, the parameter d corresponds to the size of its reference set. For FastMap, ROP, RLP and NNMap, d denotes the selected embedding dimensionality. Both the number of internal nodes of VP-trees and the number of prototypes in Zhang's method are also denoted as d. In our experiment, d is selected as 63,000, 2000, 1000, 200 and 50. For the bottom six embedding methods, there is no sense if their d is set to be 63,000. And the size of the reference set C of NNmap is set to be 5000.

Overall, NNMAP outperforms other comparable methods, and is significantly different from others. The performances of ROP and RLP are comparable with each other, and they both outperform FastMap and Brute-force. These two simple methods perform competitively. Zhang's method is comparable with FastMap, and seems to perform better at the moderate cost ($d=200$ and $d=500$). VP-trees achieve the worst trade-off between accuracy and efficiency compared with other methods. Under $d=50$, the error rate of VP-tree is high to 39.1%. It achieves the comparable error rate (0.64%) at the cost of 20,000 expensive distance computations. We note that BoostNN achieves the lowest error rates when d is relatively smaller. However, since it performs Brute-force learner in each boosting iteration, the time cost of BoostNN is significantly larger than NNMap.

The comparison of running time of these methods is reported in Table 3. This table lists the response time of the compared methods with $d=1000$ and the Brute-force with $d=63,000$. We find that NNMap has a relatively smaller time cost. In particular, compared with the baseline Brute-force, NNMap is more efficient. The error rate 0.56% is achieved by NNMap at $d=1000$ (63.2 s for each query). To achieve the comparable recognition rate, Brute-force costs 64,073 s for each query, which is nearly 64 times as that of NNMap.

6.3.2. Experiments on the CMU PIE dataset

Table 4 displays the averaged error rates attained on the CMU PIE dataset. The basic idea of Boosted Discriminant Projections (BDP) [62] and Learning Discriminant Projections and Prototypes (LDPP) [63] is also to combine multiple 1-D embeddings to a final d-D embedding. Particularly, BDP also adopts the Boosting framework in its learning procedure. However, BDP and LDPP could be applied only for a vector space, rather than for complicated data structures. In this experiment, a face image is unfolded into a vector. The k-nearest neighbor classification is also performed by using $k=1$. And the size of the reference set C of NNMap is set to be 5000. All methods are performed under varied parameters with $d=11,550, 4760, 3400, 1360, 680$ and 204.

The case on this dataset is very similar to that of the MNIST dataset. The performance is promoted with the increase of the parameter d. NNMap costs 5000 expensive distance computations to achieve the lowest error rate 4.1%, while Brute-force is required 11,550 distance computations to reach the comparable result (5.2%). VP-trees perform still worse under few prototypes compared with other methods. BDP and LDPP are very competitive, and outperform most of the methods. Particularly, under $d=204$, the recognition rate of LDPP (28.8%) is better than NNMap (30.2%). BoostNN achieves better error rates when d is relatively smaller. For other methods, the performances are comparable with each other under all selected embedding dimensionality.

Table 2
Comparison of averaged error rates (%) $\pm$ standard deviations (%) on MNIST dataset.

Method	t-test	$d=63,000$	$d=20,000$	$d=1000$	$d=500$	$d=200$	$d=50$
Brute-force	A	0.59 ± 0.16	0.57 ± 0.12	3.09 ± 0.24	4.75 ± 0.31	9.14 ± 1.42	15.36 ± 1.06
3-NN	A	0.53 ± 0.18	0.56 ± 0.15	2.91 ± 0.21	4.42 ± 0.28	8.65 ± 0.82	15.31 ± 0.93
5-NN	A	0.57 ± 0.16	0.56 ± 0.12	2.68 ± 0.26	4.16 ± 0.33	8.31 ± 0.74	14.78 ± 1.10
FastMap [52]	B	–	0.82 ± 0.16	1.18 ± 0.22	1.31 ± 0.29	1.62 ± 0.35	3.52 ± 0.46
ROP	CF	–	0.57 ± 0.20	0.61 ± 0.15	0.70 ± 0.26	0.97 ± 0.28	2.94 ± 0.37
RLP	C	–	0.62 ± 0.18	0.73 ± 0.17	0.88 ± 0.30	1.42 ± 0.34	2.41 ± 0.43
VP-trees [9]	D	–	0.64 ± 0.15	6.04 ± 0.24	7.83 ± 0.35	13.3 ± 0.41	19.0 ± 0.86
LSH [26]	E	–	0.61 ± 0.13	4.33 ± 0.21	6.25 ± 0.28	11.5 ± 0.32	16.2 ± 1.3
Zhang [7]	BF	–	0.80 ± 0.09	1.25 ± 0.19	0.73 ± 0.23	1.15 ± 0.24	2.64 ± 0.45
BoostNN	G	–	0.62 ± 0.11	0.58 ± 0.20	0.70 ± 0.25	0.67 ± 0.33	1.39 ± 0.41
NNMap	G	–	$\mathbf{0.55 \pm 0.16}$	$\mathbf{0.56 \pm 0.17}$	$\mathbf{0.67 \pm 0.20}$	$\mathbf{0.72 \pm 0.29}$	$\mathbf{1.63 \pm 0.38}$

6.3.3. Experiments on the UCI machine learning datasets

The experimental results on UCI datasets are summarized in Table 5. Both the kd-trees [10] and BBD-trees [21] are approximate nearest neighbor schemes, and expedite nearest neighbor classification. However, they could only handle vector samples. In this experiment, these two techniques are also included. The parameter d denotes the number of distance computations for each query. For all comparison methods, the tuning of the parameter d results in a wide variance of learner performance, and optimal d's of datasets are various. Generally, the larger d brings the lower error rates and the slower query respondence. To make a balance between accuracy and executive time, and simplify the comparison configuration, we set $d = 0.5 \cdot N$ uniformly. And here, $k = 1$ for nearest neighbor classification.

From Table 5, NNMap achieves the lower error rates on dataset with various sizes. Particularly, although the size of the KDD Cup 99 dataset is so large and over four million, NNMap obtains the relatively lower error rates with a small deviation. We find that BDP and LDPP are also competitive, and they perform similar to NNMap. It conforms that the parameterized models have more

flexibility for dataset. The indexing-based methods make more errors in most cases. Although the group of indexing structures could expedite the NN classifier, its accuracy is heavily limited by the tree size. When the reference set becomes small, its performance degrades seriously. It is worth noting that the simple schemes ROP and RLP achieve acceptable performance. BoostNN achieves worse error rates than NNMap when the size of datasets is larger.

6.4. Discussion

The parameters to be tuned in the proposed NNMap include d (the dimension number of the embedding space), β (the size of reference object set C), and k' (the number of nearest neighbors for computing positive ratio r^+ and negative ratio r^-). In this work, all the parameters are tuned through the three-run of five-fold cross-validation on the training data.

The influence of d to the prediction accuracy is shown in Fig. 3. We note that the averaged error rates of all the compared methods are decreasing with the increase of d. In practice, for a large-scale database, we need to find the balance between the accuracy and the efficiency. So, we need an appropriate d to achieve acceptable accuracy and efficiency. Furthermore, we find that for NNMap, the prediction tends to a stable certain performance when the d is becoming relatively large. When d is larger than a specific value, the increase of d brings a little improvement of prediction accuracy but result in a significant increase of time cost. So, we suggest that the d can be as small as possible under an acceptable prediction accuracy. From another point of view, we can set d to be as large as possible under an acceptable time cost. So we can come to the guideline that the optimized value of d is determined by trading off between the computation complexity and prediction

Table 3
Comparison of running time (s) of different methods for each query on MNIST dataset. 1-NN.

1-NN[a]	1-NN[b]	FastMap	ROP	RLP	VP-tree	LSH	Zhang	BoostNN	NNMap
4073	65.3	147.5	63.1	131.8	211.2	115.7	169.3	3978 s	63.2

[a] The Brute-force with $d = 63,000$ and 1-NN.
[b] The Brute-force with $d = 1000$.

Table 4
Comparison of averaged error rates (%) $\pm$ standard deviations (%) on CMU PIE dataset.

Method	t-test	$d = 11{,}550$	$d = 70 \times 68$	$d = 50 \times 68$	$d = 20 \times 68$	$d = 10 \times 68$	$d = 3 \times 68$
Brute-force	A	5.20 ± 1.2	13.6 ± 2.3	18.3 ± 2.5	27.3 ± 3.0	29.2 ± 3.6	42.8 ± 4.5
3-NN	A	4.93 ± 1.7	13.2 ± 1.8	17.4 ± 2.2	26.0 ± 2.8	28.5 ± 3.5	41.9 ± 3.2
5-NN	A	4.52 ± 1.1	12.7 ± 2.0	17.7 ± 1.9	26.8 ± 2.4	27.6 ± 3.9	41.3 ± 4.1
FastMap [52]	B	–	9.51 ± 1.4	14.4 ± 1.6	24.7 ± 2.5	23.6 ± 2.9	34.2 ± 4.0
ROP	C	–	8.80 ± 1.4	12.5 ± 0.8	22.1 ± 2.2	26.4 ± 3.0	36.3 ± 4.4
RLP	C	–	8.39 ± 1.8	11.7 ± 1.3	23.9 ± 2.6	25.2 ± 3.5	33.7 ± 3.8
VP-trees [9]	D	–	14.1 ± 1.4	19.1 ± 1.8	30.6 ± 2.9	34.7 ± 3.8	44.1 ± 4.3
LSH [26]	D	–	16.8 ± 1.2	18.7 ± 1.5	28.2 ± 3.3	30.9 ± 3.2	43.5 ± 4.6
BDP [62]	E	–	6.74 ± 0.7	10.8 ± 0.9	20.6 ± 2.1	23.1 ± 3.3	31.9 ± 4.0
LDPP [63]	F	–	5.60 ± 0.8	10.5 ± 1.4	17.2 ± 2.7	19.5 ± 2.4	28.8 ± 3.7
BoostNN	G	–	4.84 ± 1.6	8.55 ± 1.8	12.6 ± 2.9	15.1 ± 3.1	28.4 ± 3.8
NNMap	H	–	$\mathbf{4.13 \pm 1.5}$	$\mathbf{8.26 \pm 1.2}$	$\mathbf{13.8 \pm 2.4}$	$\mathbf{16.0 \pm 2.7}$	$\mathbf{30.2 \pm 4.1}$

Table 5
Comparison of averaged error rates (%) $\pm$ standard deviations (%) on UCI machine learning datasets.

Method	t-test	Vehicle	Spambase	Satimage	Adult	Census-inc	Covertype	PokerHand	KDDCup99
Brute-force	A	29.6 ± 3.7	17.2 ± 1.5	12.8 ± 1.8	24.9 ± 2.9	10.5 ± 2.1	6.21 ± 0.3	43.7 ± 5.6	0.31 ± 0.04
3-NN	A	28.8 ± 3.8	17.5 ± 1.8	13.4 ± 1.5	24.3 ± 2.2	10.1 ± 2.3	5.98 ± 1.1	42.2 ± 4.3	0.27 ± 0.09
5-NN	A	29.2 ± 4.2	16.8 ± 1.5	12.1 ± 2.1	24.2 ± 2.6	9.7 ± 1.8	6.03 ± 0.7	40.8 ± 5.4	0.30 ± 0.11
FastMap [52]	BC	30.4 ± 3.2	15.0 ± 1.2	8.13 ± 1.3	23.6 ± 2.3	8.17 ± 1.6	6.43 ± 1.4	37.4 ± 4.2	0.26 ± 0.07
ROP	C	28.3 ± 4.1	15.2 ± 1.4	8.59 ± 1.5	24.5 ± 2.7	8.42 ± 1.3	5.05 ± 0.8	31.1 ± 3.7	0.27 ± 0.06
RLP	B	30.9 ± 3.5	14.6 ± 2.1	10.6 ± 1.7	21.6 ± 1.9	9.30 ± 2.0	5.62 ± 0.4	36.5 ± 3.9	0.19 ± 0.03
VP-trees [9]	D	32.8 ± 4.2	19.4 ± 1.1	17.5 ± 2.2	27.4 ± 3.5	14.1 ± 2.5	8.24 ± 1.1	48.2 ± 4.4	0.38 ± 0.05
kd-trees [10]	E	35.2 ± 3.6	22.7 ± 2.3	19.2 ± 1.9	37.0 ± 3.1	13.7 ± 1.8	9.04 ± 1.5	46.7 ± 4.7	0.42 ± 0.10
BBD-trees [21]	D	30.9 ± 2.7	$21.0v1.8$	18.1 ± 1.6	25.8 ± 2.1	$13.2v2.2$	8.05 ± 0.7	45.8 ± 3.6	0.39 ± 0.08
LSH [26]	F	33.5 ± 3.6	19.2 ± 1.5	17.3 ± 1.2	23.9 ± 3.2	12.6 ± 2.6	7.37 ± 0.6	45.0 ± 3.2	0.35 ± 0.70
BDP [62]	G	27.9 ± 3.3	13.7 ± 1.8	8.05 ± 1.4	21.7 ± 3.0	8.33 ± 1.9	4.94 ± 0.2	38.3 ± 3.9	0.20 ± 0.02
LDPP [63]	G	27.5 ± 2.8	15.2 ± 1.4	6.72 ± 0.9	21.2 ± 2.1	7.25 ± 1.7	5.20 ± 0.6	35.6 ± 4.1	0.22 ± 0.03
BoostNN	H	26.2 ± 3.7	12.5 ± 2.2	6.19 ± 0.9	18.6 ± 3.0	7.51 ± 2.5	3.72 ± 0.7	36.5 ± 3.5	0.24 ± 0.08
NNMap	I	$\mathbf{26.8 \pm 3.4}$	$\mathbf{13.1 \pm 1.7}$	$\mathbf{6.40 \pm 1.2}$	$\mathbf{19.5 \pm 2.6}$	$\mathbf{7.83 \pm 2.4}$	$\mathbf{3.18 \pm 0.5}$	$\mathbf{34.2 \pm 3.8}$	$\mathbf{0.15 \pm 0.03}$

accuracy. In addition, in our experiments, we find that some of the combination coefficients α's are zero or near zero, which implies that the corresponding 1-D embeddings have worse qualities. We believe that optimal subset extraction from weak 1-D embeddings is a promising piece of future work.

Naturally, β should be larger than the embedding dimensionality number d. The smaller β results in the smaller d. The tuning of β closely related to the tuning of d. As shown in the experimental results, the accuracy and efficiency of NNMap mainly depends on d. On the other hand, since the boosting algorithm requires weak learners just a little better than the random guess, there is no need to prepare an extraordinarily larger 1-D embedding candidate set. So, in NNMap, we suggest that β is set to be a little larger than d.

We consider the two neighborhoods, i.e., the parameter k' for computing positive ratio and negative ratio, and the parameter k for final classification. It is necessary to explore the behavior of NNMap under various (k, k') pairs. NNMap is performed under various parameter pairs on the MNIST, CMU PIE, and the largest one of UCI datasets. Here, $d=20{,}000$ for MNIST, $d=4760$ for CMU PIE, and $d=2{,}449{,}215$ for KDD Cup 99. The behavior of NNMap is represented in Fig. 4. Each tiny square corresponds to one (k, k') pair, under which the error rate is shown in that tiny square. The background of the tiny square is gray, if its error rate is less than the mean error. From Fig. 4, we find that more tiny squares on the

top left of the box are covered by gray color. We could conclude that NNMap performs better under $k' > k$. This is agreed with our above neighborhood homogeneous assumption. The local structure of 1-D embedding should be optimized in a larger neighborhood.

7. Conclusion

In this work, we focus on the quantitative criterion of embedding quality for NN classification. Following the proposed embedding construction algorithm NNMap, a good multi-dimension embedding could be constructed. And its quality is guaranteed quantitatively. The NN classifier is performed on the embedded space with a weighted Manhattan distance. Furthermore, the embedding construction has no specific requirement for data structure and distance measure. In particular, for those applications with non-vector data structure and complicated distance measure, the NN classifier could be speed up and efficient. The experimental results suggest that the quality of NNMap embedding outperforms other alternative methods on the 10 datasets. The trade-off of the quantity of 1-D embeddings and their quality, and the generation of the reference set are still open problems. For multi-class problems, as analysis above, the requirement for the qualities of 1-D embeddings is much stronger than that of binary
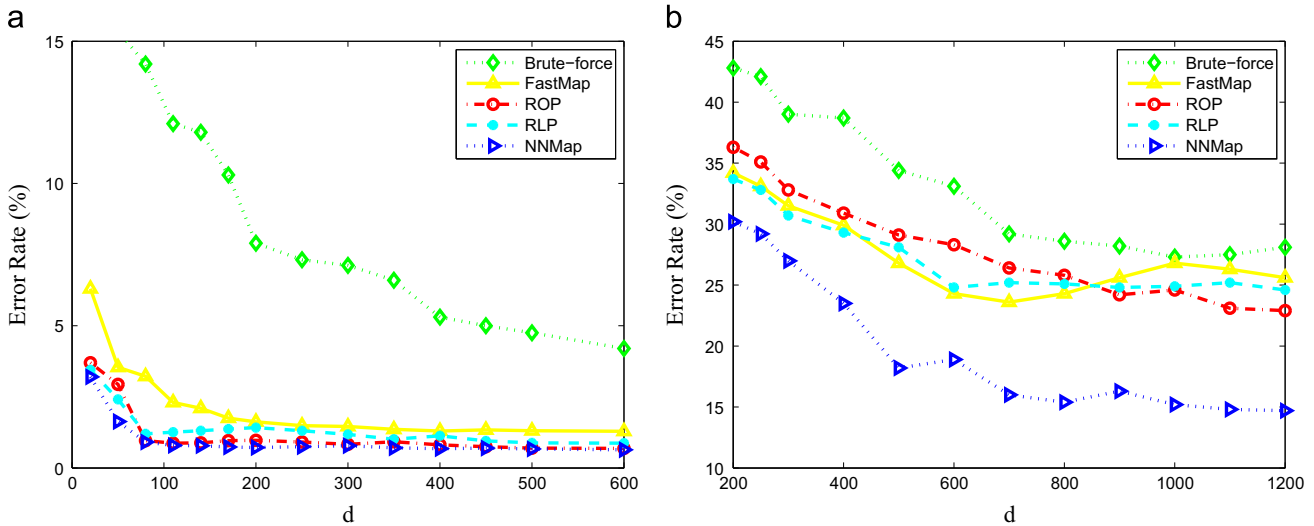

Fig. 3. The effect of d to the compared methods on the MNIST database (a) and the CMU PIE database (b).
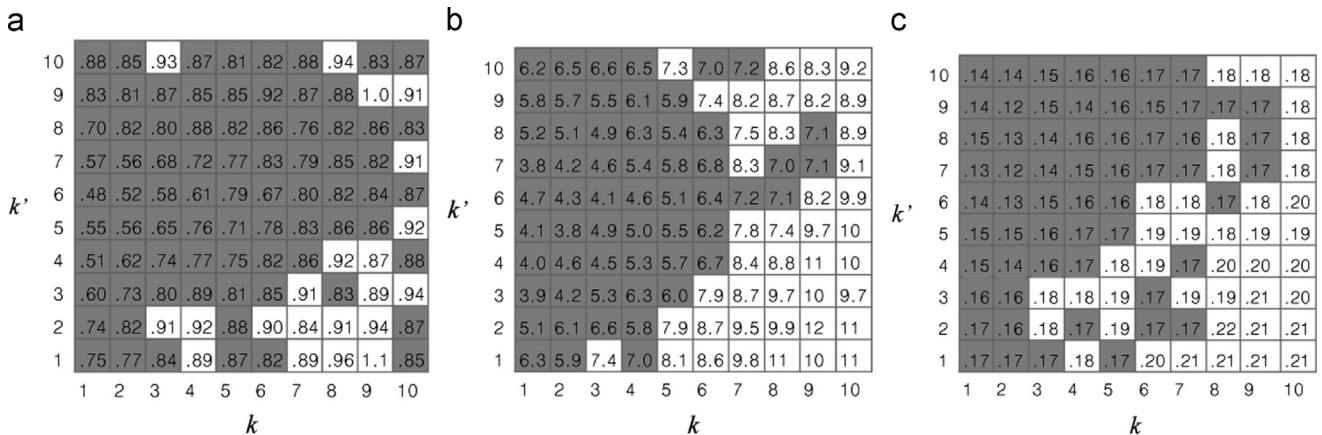

Fig. 4. The behavior of NNMap under various (k, k') pairs.

cases. The design of the alternative strategy to extend the binary case to the multi-class case is another possible future work.

Acknowledgment

References

[1] T.M. Cover, P.E. Hart, Nearest neighbor pattern classification, IEEE Trans. Inf. Theory 13 (1) (1967) 21–27.

[2] J. Wang, C. Lin, Y.C. Yang, A k-nearest-neighbor classifier with heart rate variability feature-based transformation algorithm for driving stress recognition, Neurocomputing 116 (2013) 136–143.

[3] H. Zhang, A.C. Berg, M. Maire, J. Malik SVM-KNN: discriminative nearest neighbor classification for visual category recognition, in: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2006.

[4] S. Wang, Q. Huang, S. Jiang, Q. Tian, L. Qin, Nearest-neighbor method using multiple neighborhood similarities for social media data mining, Neurocomputing 95 (2012) 105–116.

[5] O. Kucuktunc, H. Ferhatosmanoglu, λ-diverse nearest neighbors browsing for multidimensional data, IEEE Trans. Knowl. Data Eng. 25 (3) (2013) 481–493.

[6] D. Lunga, O. Ersoy, Spherical Nearest Neighbor Classification: Application to Hyperspectral Data, Purdue University. ECE Technical Report, TR-ECE-10-09 December 10, 2010.

[7] H. Zhang, J. Malik, Learning a discriminative classifier using shape context distances, in: Proceedings of Eighth IEEE International Conference on Computer Vision and Pattern Recognition, vol. 1, 2003, pp. 242–247.

[8] Y. Chen, K. Chen, M.A. Nascimento, Effective and efficient shape-based pattern detection over streaming time series, IEEE Trans. Knowl. Data Eng. 24 (February (2)) (2012) 265–278.

[9] P. Yianilos, Data structures and algorithms for nearest neighbor search in general metric spaces, in: Proceedings of ACM-SIAM Symposium on Discrete Algorithms, 1993, pp. 311–321.

[10] P. Grother, G.T. Candela, J.L. Blue, Fast implementations of nearest-neighbor classifier, Pattern Recognit. 30 (3) (1997) 459–465.

[11] M. Turk, A. Pentland, Eigenfaces for recognition, J. Cogn. Neurosci. 3 (1) (1991) 71–86.

[12] X. He, P. Niyogi, Locality preserving projections, Adv. Neural Inf. Process. Syst. 16 (2004) 153–160.

[13] E. Xing, A. Ng, M. Jordan, S. Russell, Distance metric learning with application to clustering with side-information, in: Advances in Neural Information Processing Systems, MIT Press, Cambridge, MA, 2003, pp. 505–512.

[14] J. Goldberger, S. Roweis, G. Hinton, R. Salakhutdinov, Neighbourhood Components Analysis, in: Advances in Neural Information Processing Systems, MIT Press, Cambridge, MA, 2003, pp. 513–520.

[15] P. Belhumeur, J. Hespanha, D. Kriegman, Eigenfaces vs. Fisherfaces: recognition using class specific linear projection, IEEE Trans. Pattern Anal. Mach. Intell. 19 (July (7)) (1997) 711–720.

[16] S. Yan, D. Xu, B. Zhang, H.J. Zhang, Q. Yang, S. Lin, Graph embedding and extensions: a general framework for dimensionality reduction, IEEE Trans. Pattern Anal. Mach. Intell. 29 (January (1)) (2007) 40–51.

[17] K. Stamos, N.A. Laskaris, A. Vakali, Mani-Web: large-scale web graph embedding via laplacian eigenmap approximation, IEEE Trans. Syst. Man Cybern. C Appl. Rev. 41 (9) (2011) 1–10.

[18] V. Athitsos, J. Alon, S. Sclaroff, G. Kollios, BoostMap: an embedding method for efficient nearest neighbor retrieval, IEEE Trans. Pattern Anal. Mach. Intell. 30 (January (1)) (2008).

[19] T. Denoeux, A k-nearest neighbor classification rule based on Dempster-Shafer theory, IEEE Trans. Syst. Man Cybern. 25 (5) (1995) 804–813.

[20] Z. Liu, Q. Pan, J. Dezert, Classification of uncertain and imprecise data based on evidence theory, Neurocomputing 133 (2014) 459–470.

[21] S. Arya, D.M. Mount, N.S. Netanyahu, R. Silverman, A.Y. Wu, An optimal algorithm for approximate nearest neighbor searching fixed dimensions, J. ACM 45 (6) (1998) 891–923.

[22] S. Berchtold, D.A. Keim, H.P. Kriegel. The x-tree: an index structure for high-dimensional data, in: Proceedings of VLDB '96, 1996, pp. 28–39.

[23] A. Guttman, R-trees: a dynamic index structure for spatial searching, in: Proceedings of SIGMOD '84, ACM, Boston, Massachusetts, 1984, pp. 47–57.

[24] A. Beygelzimer, S. Kakade, J. Langford, Cover trees for nearest neighbor, in: Proceedings of ICML '06, ACM, Pittsburgh, Pennsylvania, 2006, 97–104.

[25] Y. Hwang, B. Han, H.-K. Ahn, A fast nearest neighbor search algorithm by nonlinear embedding, in: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2006, pp. 3053–3060.

[26] A. Andoni, P. Indyk, Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions, Commun. ACM 51 (1) (2008) 117–122.

[27] Y. Weiss, A. Torralba, R. Fergus, Spectral hashing, in: Proceedings of NIPS '08, 2008, pp. 1753–1760.

[28] S. Korman, S. Avidan, Coherency sensitive hashing, in: Proceedings of ICCV '11 IEEE, 2011.

[29] W.H. Al-Arashi, H. Ibrahim, S.A. Suandi, Optimizing principal component analysis performance for face recognition using genetic algorithm, Neurocomputing 128 (2014) 415–420.

[30] D.D. Lee, H.S. Seung, Learning the parts of objects by non-negative matrix factorization, Nature 401 (6755) (1999) 788–791.

[31] B. Scholkopf, A. Smola, K.R. Muller, Kernel principal component analysis, in: B. Scholkopf, C.J.C. Burges, A.J. Smola (Eds.), Advance in Kernel Methods, Support Vector Learning, MIT Press, Cambridge, MA, 1999, pp. 327–352.

[32] D. Zhang, Z.H. Zhou, S. Chen, Non-negative matrix factorization on kernels, in: Proceedings of Pacific Rim International Conference on Artificial Intelligence, 2006, pp. 404–412.

[33] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, K. R. Mullers, Fisher discriminant analysis with kernels, in: Proceedings of IEEE Workshop on Neural Networks for Signal Processing, 1999, pp. 41–48.

[34] F. Dinuzzo, Learning output kernels for multi-task problems, Neurocomputing 118 (2013) 119–126.

[35] S. Roweis, L. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (5500) (2000) 2323–2326.

[36] J.B. Tenenbaum, V. de Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (5500) (2000) 2319–2323.

[37] M. Belkin, P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, Neural Comput. 15 (2003) 1373–1396.

[38] N. Shental, T. Hertz, D. Weinshall, M. Pavel, Adjustment learning and relevant component analysis, in: Proceedings of the Seventh ECCV, vol. 4, Copenhagen, Denmark, 2002, pp. 776–792.

[39] S.C.H. Hoi, W. Liu, M.R. Lyu, W.Y. Ma, Learning distance metric with contextual constraints for image retrieval, in: Proceedings of the 11th IEEE International Conference on Computer Vision and Pattern Recognition, 2006.

[40] T. Hastie, R. Tibshirani, Discriminant adaptive nearest neighbor classification, IEEE Trans. Pattern Anal. Mach. Intell. 18 (June (6)) (1996) 607–616.

[41] K. Weinberger, J. Blitzer, L. Saul, Distance metric learning for large margin nearest neighbor classification, in: Advances in Neural Information Processing Systems, MIT Press, Cambridge, MA, 2006.

[42] L. Yang, R. Jin, R. Sukthankar, Y. Liu, An efficient algorithm for local distance metric learning, in: Proceedings of AAAI Conference on Artificial Intelligence, Boston, MA, 2006, pp. 543–548.

[43] A. Frome, Y. Singer, J. Malik, Image retrieval and classification using local distance functions, in: Advances in Neural Information Processing Systems, MIT Press, Cambridge, MA, 2007.

[44] R.E. Schapire, The boosting approach to machine learning: an overview, in: MSRI Workshop on Nonlinear Estimation and Classification, 2002.

[45] L. Breiman, Bagging predictors, Mach. Learn. 24 (1996) 123–140.

[46] E. Bauer, R. Kohavi, An empirical comparison of voting classification algorithms: bagging, boosting, and variants, Mach. Learn. 36 (July (1)) (1999) 105–139.

[47] Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, J. Comput. Syst. Sci. 55 (1997) 119–139.

[48] T.G. Dietterich, An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization, Mach. Learn. 40 (August (2)) (2000) 139–157.

[49] H. Schwenk, Y. Bengio, Boosting neural networks, Neural Comput. 12 (2000) 1869–1887.

[50] H. Drucker, Improving regressor using boosting, in: Proceedings of the 14th International Conference on Machine Learning, Morgan Kaufmann, San Francisco, CA, 1997, pp. 107–115.

[51] J. Bourgain, On Lipschitz embedding of finite metric spaces in hilbert space, Isr. J. Math. 52 (1985) 46–52.

[52] C. Faloutsos, K. Lin, FastMap: a fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets, in: Proceedings of SIGMOD, International Conference on Management of Data, vol. 24, no. 2, June 1995, pp. 163–174.

[53] J.T.L. Wang, X. Wang, D. Shasha, K. Zhang, MetricMap: an embedding technique for processing distance-based queries in metric spaces, IEEE Trans. Syst. Man Cybern. B Cybern. 35 (5) (2005) 973–987.

[54] G. Hristescu, M. Farach-Colton, Cluster-Preserving Embedding of Proteins, Technical Report 99-50, Computer Science Department, Rutgers University, 1999.

[55] T.-K. Kim, J. Kittler, Locally linear discriminant analysis for multimodally distributed classes for face recognition with a single model image, IEEE Trans. Pattern Anal. Mach. Intell. 27 (3) (2005) 318–327.

[56] Z. Fan, Y. Xu, D. Zhang, Local linear discriminant analysis framework using sample neighbors, IEEE Trans. Neural Netw. 22 (7) (2011) 1119–1132.

[57] J. Vleugels, R.C. Veltkamp, Efficient image retrieval through vantage objects, Pattern Recognit. 35 (January (1)) (2002) 69–80.

[58] M.F. Balcan, A. Blum, N. Srebro, A theory of learning with similarity functions, Mach. Learn. 72 (2008) 89–112.
[59] S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape contexts, IEEE Trans. Pattern Anal. Mach. Intell. 24 (April (4)) (2002) 509–522.
[60] S.C. Sahinalp, M. Tasan, J. Macker, Z.M. Ozsoyoglu, Distance based indexing for string proximity search, in: Proceedings of IEEE International Conference on Data Engineering, 2003, pp. 125–136.
[61] Y.D. Mu, S.C. Yan, Non-metric locality-sensitive hashing, in: Proceedings of the 24th AAAI Conference on Artificial Intelligence, 2003, pp. 125–136.
[62] D. Masip, J. Vitrià, Boosted discriminant projections for nearest neighbor classification, Pattern Recognit. 39 (2006) 164–170.
[63] M. Villegas, R. Paredes, Simultaneous learning of a discriminative projection and prototypes for nearest-neighbor classification, in: Proceedings of the 13th IEEE International Conference on Computer Vision and Pattern Recognition, 2008.

Jing Chen received the B.S. degree in Computational Mathematics from Shandong University, Ji'nan, China, the M.S. and Ph.D. degrees in Computer Science and Technology from Chongqing University, Chongqing, China. He is currently a Post-Doctor Fellow with the Faculty of Science and Technology, University of Macau, Macau. His research interests include machine learning, pattern recognition, and biomedical information processing.

Yuan Yan Tang received the B.S. degree in electrical and computer engineering from Chongqing University, Chongqing, China, the M.S. degree in electrical engineering from the Beijing University of Post and Telecommunications, Beijing, China, and the Ph.D. degree in computer science from Concordia University, Montreal, Quebec, Canada. He is a Chair Professor in the Faculty of Science and Technology at the University of Macau (UM). Before joining UM, he served as a Chair Professor in the Department of Computer Science at Hong Kong Baptist University and dean of the College of Computer Science at Chongqing University, China. He is a Chair of the Technical Committee on Pattern Recognition of the IEEE Systems, Man, and Cybernetics Society (IEEE SMC) for his great contributions to wavelet analysis, pattern recognition, and document analysis. Recently, he was elected as one of the executive directors of the Chinese Association of Automation Council. With all his distinguished achievement, he is also the Founder and Editor-in-Chief of the International Journal of Wavelets, Multi-resolution, and Information Processing (IJWMIP), and an Associate Editor of the International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI), and the International Journal on Frontiers of Computer Science (IJFCS). He has been presented with numerous awards such as the First Class of Natural Science Award of Technology Development Centre, Ministry of Education of the People's Republic of China, in November 2005, and the Outstanding Contribution Award by the IEEE Systems, Man, and Cybernetics Society, in 2007. He has published more than 300 papers, books, and book chapters. He is a Fellow of the IEEE and of the Pattern Recognition Society (IAPR).

C.L. Philip Chen received the M.S. degree in electrical engineering from the University of Michigan, Ann Arbor, in 1985, and the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN, in 1988. From 1989 to 2002, he was with the Department of Computer Science and Engineering, Wright State University, Dayton, OH, as an Assistant, an Associate, and a Full Professor before he joined the University of Texas, San Antonio, where he was a Professor and the Chair of the Department of Electrical and Computer Engineering and the Associate Dean for Research and Graduate Studies of the College of Engineering. He is currently a Chair Professor of the Department of Computer and Information Science and Dean of the Faculty of Science and Technology, University of Macau, Macau, China. He is a Fellow of the AAAS. He is currently the President of the IEEE Systems, Man, and Cybernetics Society. In addition, he is an Accreditation Board of Engineering and Technology Education Program Evaluator for Computer Engineering, Electrical Engineering, and Software Engineering programs.

Bin Fang received the B.S. degree in electrical engineering from Xi'an Jiaotong University, Xi'an, China, the M.S. degree in electrical engineering from Sichuan University, Chengdu, China, and the Ph.D. degree in electrical engineering from the University of Hong Kong, Hong Kong. He is currently a Professor with the Department of Computer Science, Chongqing University, Chongqing, China. His research interests include computer vision, pattern recognition, medical image processing, biometrics applications, and document analysis. He has published more than 100 technical papers and is an Associate Editor of the International Journal of Pattern Recognition and Artificial Intelligence.

Zhaowei Shang received the B.S. degree in computer science from the Northwest Normal University, Lanzhou, China, in 1991, the M.S. degree from the Northwest Polytechnical University, Xi'an, China, in 1999, and the Ph.D. degree in computer engineering from Xi'an Jiaotong University, Xi'an, in 2005. He is currently a Professor with the Department of Computer Science, Chongqing University, Chongqing, China. His research interests include computer vision, pattern recognition, image processing, and wavelet analysis.

Yuewei Lin received the B.S. degree in Optical Information Science & Technology from Sichuan University, Chengdu, China, and the M.E. degree in Optical Engineering from Chongqing University, Chongqing, China. He is currently working toward the Ph.D. degree in the Department of Computer Science & Engineering, University of South Carolina. His current research interests include computer vision, machine learning and image/video processing.