Manifold based local classifiers: linear and nonlinear approaches

Hakan Cevikalp, Diane Larlus, Mike Neamtu, William Triggs, Frédéric Jurie

J Sign Process Syst
DOI 10.1007/s11265-008-0313-4

Manifold Based Local Classifiers: Linear and Nonlinear Approaches

Hakan Cevikalp · Diane Larlus · Marian Neamtu · Bill Triggs · Frederic Jurie

Received: 12 February 2008 / Revised: 22 September 2008 / Accepted: 27 October 2008

Abstract In case of insufficient data samples in high-dimensional classification problems, sparse scatters of samples tend to have many 'holes'—regions that have few or no nearby training samples from the class. When such regions lie close to inter-class boundaries, the nearest neighbors of a query may lie in the wrong class, thus leading to errors in the Nearest Neighbor classification rule. The K-local hyperplane distance nearest neighbor (HKNN) algorithm tackles this problem by approximating each class with a smooth nonlinear manifold, which is considered to be locally linear. The method takes advantage of the local linearity assumption by using the distances from a query sample to the affine hulls of query's nearest neighbors for decision making. However, HKNN is limited to using the Euclidean distance metric, which is a significant limitation in practice. In this paper we reformulate HKNN in terms of subspaces, and propose a variant, the Local Discriminative Common Vector (LDCV) method, that is more suitable for classification tasks where the classes have similar intra-class variations. We then extend both methods to the nonlinear case by mapping the nearest neighbors into a higher-dimensional space where the linear manifolds are constructed. This procedure allows us to use a wide variety of distance functions in the process, while computing distances between the query sample and the nonlinear manifolds remains straightforward owing to the linear nature of the manifolds in the mapped space. We tested the proposed methods on several classification tasks, obtaining better results than both the Support Vector Machines (SVMs) and their local counterpart SVM-KNN on the USPS and Image segmentation databases, and outperforming the local SVM-KNN on the Caltech visual recognition database.

Keywords Affine hull · Common vector · Convex hull · Distance learning · Image categorization · Local classifier · Manifold learning · Object recognition

H. Cevikalp (✉)
Electrical and Electronics Engineering Department,
Eskisehir Osmangazi University,
Meselik,
26480 Eskisehir, Turkey
e-mail: hakan.cevikalp@gmail.com

D. Larlus
Learning and Recognition in Vision (LEAR), INRIA,
Grenoble, France
e-mail: diane.larlus@inrialpes.fr

M. Neamtu
Department of Mathematics, Vanderbilt University,
Nashville, TN, USA
e-mail: neamtu@math.vanderbilt.edu

B. Triggs
Laboratoire Jean Kuntzmann,
Grenoble, France
e-mail: Bill.Triggs@imag.fr

F. Jurie
University of Caen,
Caen, France
e-mail: frederic.jurie@unicaen.fr

1 Introduction

Despite its age and simplicity, the Nearest Neighbor (NN) classification rule—assigning the query sample to the class with the closest training sample—is among the most successful and robust methods for many classification problems. Various distance functions can be used to measure the proximity including the Euclidean and Mahalanobis distances. It has been shown theoretically that NN classification has good asymptotic performance committing at most twice as

Published online: 07 December 2008

many errors as the optimal Bayes rule classifier. Empirically, NN classifiers with well-chosen distance metrics outperform more sophisticated classifiers in many situations [1–3].

Unfortunately, the NN algorithm does not have good generalization ability when there are only a limited number of examples in high-dimensional spaces: *Hole artifacts* occur in the decision surface owing to random variations in sampling density, and this reduces the generalization performance [2–4]. Various methods have been proposed in the literature to overcome this pitfall [2–6]. In [3], the authors perform a local linear discriminant analysis on the nearest neighbors to deform the Euclidean distance. However, this method is not suitable for high-dimensional classification tasks as there are not enough nearest neighbors for a linear discriminant analysis. Peng et al. [7] proposed the Adaptive Quasiconformal Kernel Nearest Neighbors algorithm which warps the input space based on the local posterior probability estimates and weighted Mahalanobis distance. However, the class covariance matrices become rank deficient in high-dimensional spaces when there is a limited amount of data. As a result, the Mahalanobis distance cannot be computed in these situations, which in turn limits application areas of the method. In [8], the authors introduced the ADAMENN algorithm, in which the Euclidean distance is deformed in a neighborhood of the query points by means of a *local feature relevance factor*. However, the method requires tuning six parameters which can be confusing. Domeniconi and Gunopulos [5] train a global SVM classifier on the entire data and use it to deform the distances locally. In most applications of SVMs, the number of extracted support vectors is small compared to the training set size, but in high-dimensional applications most of the training examples become support vectors. As the method proposed in [5] requires online kernel evaluations for each support vector, it becomes inefficient for real-time high-dimensional classification applications. Furthermore, derivations are done for 2 classes and the method is not generalized for multi-class case, which limits its practical applicability. In [2], Peng et al. train local linear SVMs, rather than a single global nonlinear one, using the nearest samples, and weigh the distance metric based on the linear separating hyperplanes obtained from these classifiers. As before, the method is not designed for the multi-class tasks. Furthermore, besides the number of nearest neighbors, it contains two extra parameters that need to be fixed empirically and it assumes the classes to be locally linearly separable, which is not always true in practice. Similarly, Zhang et al. proposed a local classifier for the high-dimensional case, called SVM-KNN [6], without deforming the distance metric. This method trains SVMs on the nearest samples using various distance functions and uses them to classify the query sample. While using various distance metrics may seem appealing, decreasing the available nearest neighbors through SVM is undesirable since the extracted support vectors may not model the local decision boundaries correctly. Furthermore, the local decision boundaries are not constructed with respect to the query sample, which may yield unreasonable assignments. Another local method, HKNN, was proposed for high-dimensional data and has been shown to work well in several classification tasks [4, 9]. In this method, each class is modeled as a smooth low-dimensional manifold embedded in the high-dimensional space. In this setting it is reasonable to assume that the manifolds are locally linear. If the training data is limited, new points can be *fantasized* to approximate each manifold locally: given a query sample, the nearest neighbors in each class are used to construct a local linear manifold for the class. The query sample is classified based on its distances to these local linear manifolds. This reduces the negative effects of the sparse training data giving significant improvements in recognition accuracies.

Linearity assumptions for local regions are also widely used for nonlinear dimensionality reduction [10–12]. Hinton et al. [10] introduced Mixtures of Principal Component Analyzers to approximate the underlying nonlinear data manifolds for classification of handwritten digits. Roweis and Saul [11] proposed Locally Linear Embedding, in which nonlinear structure of the high-dimensional data is approximated by exploiting the linear reconstructions. Similarly, Verbeek [12] combined several locally valid linear manifolds to obtain a global nonlinear mapping between the high-dimensional sample space and the low-dimensional manifold. Another application for the local linear manifolds is the identification and matching of faces. Cevikalp et al. [13] projected face images onto a linear manifold removing the within-class variations in the process. Kim and Kittler [14] used K-means clustering to partition each face class into several disjoint sets, following which linear basis vectors are extracted by treating each set as a linear manifold. Finally, locally linear manifolds are used to approximate the nonlinear global structure of each class. Fitzgibbon and Zisserman [15] used linear manifolds in the context of face matching to discover the principal cast of a movie. The many applications of linear manifolds in the context of classification can be attributed in part to their simplicity and computational efficiency. Finding distances from query samples to linear manifolds requires only simple linear algebra. On the other hand, computing distances to nonlinear manifolds can be problematic. Even if the manifolds are restricted to being convex hulls, distance computations require the solution of quadratic optimization problems. As the structure becomes more complex, computing distances to these structures becomes more difficult.

Recently, the development of specialized distance functions for classification tasks and dimensionality reduction has

emerged as a fruitful line of research. Significant improvements have been achieved using task specific distance metrics [6, 16, 17]. For instance, for Euclidean distance, HKNN achieved the best recognition rate among all of the methods discussed in [6], but it was outperformed by SVM-KNN trained using alternative distance metrics. Zhang et al. [16] reported that the Chi-square and Earth Mover's distances show significant improvements over linear kernels for image classification tasks using histogram based features. In [17], Tenenbaum proposed the Isomap method, in which geodesic distances are used for nonlinear dimensionality reduction. All of these results support the hypothesis that it is important to choose a distance metric that is well-adapted to the given application.

In this paper we propose a useful nonlinearization process that extends HKNN allowing it to use a wide variety of distances. The main idea is to map the nearest neighbor samples nonlinearly into a higher-dimensional space using a suitable distance function and then construct linear manifolds in this new space. The nearest neighbors are thus transformed into a more discriminative feature space, which in turn improves the recognition accuracy. Moreover, since the classification problem is cast in a higher-dimensional space, the classes are typically linearly separable. Thus, the local linearity assumption of class manifolds around the query point is more likely to be satisfied. The nonlinearization process also allows us to apply HKNN to new classification tasks for which direct application was not feasible. Although the constructed manifolds correspond to nonlinear structures in the original sample space, finding distances to them is still straightforward owing to their linear nature in the mapped space. In this study, we also introduce a variation of HKNN, called the Local Discriminative Common Vector (LDCV) method, which is better suited for classifications tasks where the classes have similar local intra-class variations. Then, we extend the LDCV to the nonlinear case using the same nonlinearization process.

The remainder of the paper is organized as follows: In Section 2 we reformulate HKNN in terms of subspaces and generalize it to the nonlinear case using the kernel trick. We also briefly review a related method, K-local convex distance nearest neighbor (CKNN). In Section 3, the LDCV method is introduced and kernelized. Section 4 describes the data sets and experimental results. Finally, we draw conclusions in Section 5.

2 The Nonlinearization of the HKNN Method

We first formulate HKNN in terms of subspaces, then extend it to the nonlinear case using subspace concepts and the kernel trick.

2.1 Formulation of the HKNN Method in Terms of Subspaces

In HKNN, given a query sample, the first step is to find for each class the K training points nearest to the query. These neighbors are then used to construct a local linear manifold for each class in the training set. Finally the query sample is assigned to the class associated with the closest manifold. We now formulate this process using subspaces.

Suppose there are C classes in the training set. Let $V_i^K(x_q) = \{x_1^i, x_2^i, \ldots, x_K^i\}$ denote the set of the K-nearest samples of the query sample $x_q \in \Re^d$ in the training set, belonging to the i-th class. Here we suppose that the dimension d of the sample space is larger than or equal to K, or more generally that the affine hull of the nearest neighbors from each class is a proper subset of $\Re^d$ of dimension less than d. The local affine hull of class i is defined in terms of the K-nearest neighbors as

$$\mathrm{LH}_i^K(x_q) = \left\{ p | p = \sum_{m=1}^{K} \alpha_m^i x_m^i, \quad \alpha_m^i \in \Re, \quad \sum_{m=1}^{K} \alpha_m^i = 1 \right\}, \quad i = 1, .., C. \tag{1}$$

Note that this is the lowest-dimensional linear manifold that passes through all points of $V_i^K(x_q)$. We can get rid of the constraint $\sum_{m=1}^{K} \alpha_m^i = 1$, by choosing any reference point from $V_i^K(x_q)$, e.g., the mean $\mu_i = (1/K) \sum_{m=1}^{K} x_m^i$, and rewrite Eq. 1 as:

$$\mathrm{LH}_i^K(x_q) = \left\{ p | p = \mu_i + \sum_{m=1}^{l_i} \beta_m^i z_m^i, \quad \beta_m^i \in \Re \right\}, \quad i = 1, .., C, \tag{2}$$

where the set $\{z_1^i, z_2^i, \ldots, z_{l_i}^i\}$ is any linear basis of the *difference subspace* [18] spanned by the difference vectors $\{x_1^i - \mu_i, x_2^i - \mu_i, \ldots, x_K^i - \mu_i\}$, and $l_i \leq K - 1$ is the dimension of the basis. Note that there is no constraint on the sum of the new coefficients β_m^i. The difference subspace is also the range space of the covariance matrix of the samples in $V_i^K(x_q)$ [13].

In order to classify a query point x_q, the minimum distances between the query and the local linear manifolds must be computed. Then the query is assigned to the class whose manifold is closest to x_q. The minimum distance between x_q and the i-th linear manifold is computed by:

$$d\left(x_q, \mathrm{LH}_i^K(x_q)\right) = \min_{p \in \mathrm{LH}_i^K(x_q)} \left\| x_q - p \right\| = \min_{\beta^{(i)} \in \Re^{l_i}} \left\| x_q - \mu_i - Z^{(i)} \beta^{(i)} \right\|, \quad i = 1, \ldots, C, \tag{3}$$

where $Z^{(i)}$ is a matrix whose columns are the independent difference vectors and $\beta^{(i)}$ is a column vector containing the coefficients β_m^i [4]. Here $\|\cdot\|$ denotes the Euclidean norm. Minimization of the above equation leads to:

$$\beta^{(i)} = \left(Z^{(i)T} Z^{(i)}\right)^{-1} Z^{(i)T}\left(x_q - \mu_i\right). \tag{4}$$

Note that the matrix $P^{(i)} = Z^{(i)}\left(Z^{(i)T} Z^{i}\right)^{-1} Z^{(i)T}$ defines an orthogonal projection operator. In our case, it is the orthogonal projection operator onto the difference subspace of $V_i^K(x)$. Thus, we can rewrite Eq. 3 as:

$$\begin{aligned} d\left(x_q, \mathrm{LH}_i^K\left(x_q\right)\right) &= \left\|x_q - \mu_i - P^{(i)}\left(x_q - \mu_i\right)\right\| \\ &= \left\|\left(I - P^{(i)}\right)\left(x_q - \mu_i\right)\right\|, \end{aligned} \tag{5}$$

where I is the identity matrix and $P^{(i)}$ is the orthogonal projection operator onto the difference subspace of the i-th class. The matrix $P_{\mathrm{NS}}^{(i)} = \left(I - P^{(i)}\right)$ is called the orthogonal projection operator onto the *indifference subspace* (the null space of the covariance matrix) of $V_i^K\left(x_q\right)$ [13]. Notice that the difference and indifference subspaces are orthogonal complements of each other. All points $x_{\mathrm{aff}}^i \in \mathrm{LH}_i^K\left(x_q\right)$ project to a unique vector:

$$x_{\mathrm{com}}^i = P_{\mathrm{NS}}^{(i)} x_{\mathrm{aff}}^i, \quad i = 1, \ldots, C, \tag{6}$$

that characterizes the manifold. The minimum distance of the query vector to each manifold can be written as the standard Euclidean distance between the projected vectors:

$$d\left(x_q, \mathrm{LH}_i^K\left(x_q\right)\right) = \left\|P_{\mathrm{NS}}^{(i)} x_q - x_{\mathrm{com}}^i\right\|, \; i = 1, .., C. \tag{7}$$

Thus the problem can be seen as a subspace problem where each local subspace is modeled with the associated indifference subspace of the nearest neighbors in the vicinity of the query sample. It is clear that each class is represented with a unique vector obtained removing intra-class variations among the local nearest samples in this setting. This is complementary to the Mahalanobis distance in the sense that it gives a natural distance associated with the null space of the covariance matrix, not the span. The final decision function for a given query x_q can be written as:

$$\begin{aligned} g\left(x_q\right) &= \arg\min_{i=1,\ldots,C}\left(\left\|P_{\mathrm{NS}}^{(i)} x_q - x_{\mathrm{com}}^i\right\|\right) \\ &= \arg\min_{i=1,\ldots,C}\left(\left\|P_{\mathrm{NS}}^{(i)}\left(x_q - \mu_i\right)\right\|\right). \end{aligned} \tag{8}$$

Since the projection matrices are idempotent, i.e., $\left(P_{\mathrm{NS}}^{(i)}\right)^2 = P_{\mathrm{NS}}^{(i)}$, the above classification rule yields quadratic decision boundaries around the query sample.

For computational efficiency in high dimensions, the projection operators can be represented implicitly by $P^{(i)} = Q_i Q_i^T$ and $P_{\mathrm{NS}}^{(i)} = I - Q_i Q_i^T$ where Q_i is the 'Q' matrix of the QR decomposition of the matrix of difference vectors.

2.2 The K-Local Convex Distance Nearest Neighbor (CKNN) Method

Rather than using distances to affine hulls as in HKNN, we can also use distances to convex hulls—the convex spans of the K-nearest neighbors. This corresponds to adding the constraints $\alpha_m^i \geq 0$, $m = 1, \ldots, K$, $i = 1, \ldots, C$ to Eq. 1. The resulting method is called as K-Local Convex Distance Nearest Neighbor (CKNN) [4]. In Fig. 1, we give an illustrative example comparing HKNN and CKNN methods on a two-class problem with K=2.

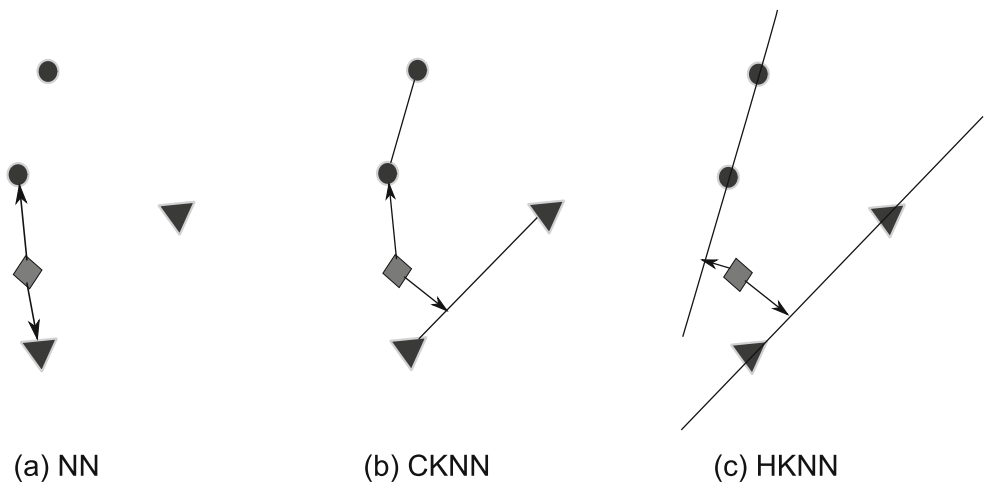

Figure 1 Comparison of NN, CKNN and HKNN methods. The closest distance from a query to an affine/convex hull is the norm of displacement from the query to the closest point on the hull. Observe how the distances change by each method.

In general, finding the distance between the query point and a convex hull requires the solution of the following quadratic programming problem:

$$\min_{\alpha^{(i)}} \frac{1}{2} \left\| x_q - X_i \alpha^{(i)} \right\|^2 \quad \text{s.t.} \sum_{m=1}^{K} \alpha_m^i = 1,\ \alpha_m^i \geq 0\ , m = 1, \ldots, K, \tag{9}$$

where X_i is the matrix whose columns are the nearest neighbors in $V_i^K(x_q)$ [19]. Once the optimal coefficient vector $\alpha^{(i)*}$ is found, $\left\| x_q - X_i \alpha^{(i)*} \right\|$ determines the distance from x_q to the local convex hull of the class i. This is repeated for each class and the query is assigned to the class whose convex hull distance is the closest. In case of affinely independent samples (i.e., corresponding difference vectors are independent, so the convex hull is a simplex), a simple intuitive method based on successive affine projections can also be applied to compute the convex distance: repeatedly compute the affine projection weights using Eq. 4, compute $\alpha^{(i)}$ from $\beta^{(i)}$ and discard the basis point with the most negative α_m^i until all weights are positive.

2.3 Kernelization Process

Before introducing the kernelization of the HKNN algorithm, we need the following definitions. The local scatter matrix S_i^K of the nearest neighbors belonging to the i-th class is defined as:

$$S_i^K = \sum_{m=1}^{K} \left(x_m^i - \mu_i\right)\left(x_m^i - \mu_i\right)^T, \quad i = 1, \ldots, C. \tag{10}$$

HKNN algorithm uses projections onto the null spaces of these matrices to classify the query sample. Similarly, the local total scatter matrix S_T^K of all nearest neighbors in the vicinity of the query sample is defined as:

$$S_T^K = \sum_{i=1}^{C} \sum_{m=1}^{K} \left(x_m^i - \mu\right)\left(x_m^i - \mu\right)^T, \tag{11}$$

where μ is the mean of all nearest neighbors.

The kernel trick can be used to map the data into a higher-dimensional space as in the Kernel Principal Component Analysis (Kernel PCA) [20] approach, but the HKNN algorithm uses the null spaces of the local covariance matrices not range spaces. Therefore, we need to modify HKNN to work in terms of dot products of the mapped samples in $\mathcal{F}$.

First it should be noted that—so long as we are only interested in differences of distances to various classes—it suffices to work within the span of the local total scatter matrix. The orthogonal space—the null space of the local total scatter—is a common null subspace of each of the individual-class local scatters, so the projection along this subspace is the same for all classes: only projections within the span of the local total scatter differ among classes [21]. The total squared distance from a query to a class is the sum of the squared distance to the span of the local total scatter (a constant for all classes) and of the squared distance to the span of the local class scatter within the range of total scatter subspace as illustrated in Fig. 2. Hence, we first project the query onto the range of the local total scatter matrix S_T^K using PCA, and then compute distances from the projection to each manifold in the projected space. This translates into using the projection matrices $P_{\text{int}}^{(i)}$, $i = 1, \ldots, C$, of the intersections of the null space of the local scatter matrices $N\left(S_i^K\right)$ with the range of the total scatter matrix $R\left(S_T^K\right)$, to compute distances. If we let P represent the projection matrix onto $R\left(S_T^K\right)$, then the projection matrix $P_{\text{int}}^{(i)}$ of the intersection $N\left(S_i^K\right) \cap R\left(S_T^K\right)$ for each class can be found as:

$$P_{int}^{(i)} = P_{NS}^{(i)} P = P P_{NS}^{(i)}, \quad i = 1, \ldots, C, \tag{12}$$

since the projection matrices of $N\left(S_i^K\right)$ and $R\left(S_T^K\right)$ commute as shown in Theorem 1 given in the Appendix. Notice that, in general, the projection matrix of any intersection cannot be obtained using Eq. 12 if the projection matrices of the associated subspaces do not commute [22].

Using intersection subspaces does not change the assignment of a query sample. In other words, the decision boundaries around the query sample x_q obtained by $P_{\text{int}}^{(i)}$ and $P_{\text{NS}}^{(i)}$, $i = 1, \ldots, C$, yield same label for x_q. This is also formally proved in Theorem 2 given in the Appendix.

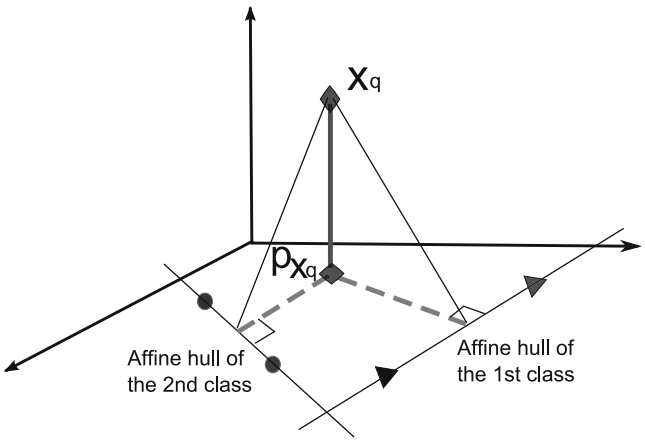

Figure 2 The modified HKNN approach employs the distances between the query sample and each manifold in the PCA projected space. These computed distances give rise to same assignment for the query sample as in the original HKNN method.

Notice that, it may not be possible to compute the distances directly between the query sample and the constructed linear manifolds in the nonlinearly mapped space since the dimensionality can be infinite. However, since we are only interested in assignment of the query to the nearest manifold, the same goal can be accomplished using the relative distances in the lower-dimensional PCA projected space as illustrated in Fig. 2.

2.4 Nonlinear HKNN (NHKNN) Method

The above procedure is convenient because it allows one to construct the projection onto the total span using dot products of samples and thus it can be used to extend the HKNN method to the nonlinear case. Let $\Phi = \left[\Phi^{(1)}, \Phi^{(2)}, \ldots, \Phi^{(C)}\right]$ represent the matrix whose columns are the mapped nearest neighbors in the implicit high-dimensional feature space, $\Im$, where $\Phi^{(i)} = \left[\phi\left(x_1^i\right), \phi\left(x_2^i\right), \ldots, \phi\left(x_K^i\right)\right]$ is the matrix whose columns are the nearest neighbors in $V_i^K\left(x_q\right)$. Suppose $M=CK$ is the total number of nearest neighbors around the query sample. The local scatter matrix S_i^Φ of each class and the scatter matrix S_T^Φ of the pooled neighbors in $\Im$ are given by:

$$\begin{aligned} S_i^\Phi &= \sum_{m=1}^{K} \left(\phi\left(x_m^i\right) - \mu_i^\Phi\right)\left(\phi\left(x_m^i\right) - \mu_i^\Phi\right)^T \\ &= \left(\Phi^{(i)} - \Phi^{(i)} 1_K\right)\left(\Phi^{(i)} - \Phi^{(i)} 1_K\right)^T, \quad i = 1, \ldots, C, \end{aligned} \tag{13}$$

$$\begin{aligned} S_T^\Phi &= \sum_{i=1}^{C} \sum_{m=1}^{K} \left(\phi\left(x_m^i\right) - \mu^\Phi\right)\left(\phi\left(x_m^i\right) - \mu^\Phi\right)^T \\ &= (\Phi - \Phi 1_M)(\Phi - \Phi 1_M)^T, \end{aligned} \tag{14}$$

where μ_i^Φ is the mean of mapped nearest neighbors in $V_i^K\left(x_q\right)$, μ^Φ is the mean of all mapped nearest neighbors in the vicinity of the query sample. Here $1_K \in \Re^{K\times K}$ is a matrix whose elements are all $1/K$ and $1_M \in \Re^{M\times M}$ is a matrix with entries $1/M$. The kernel matrix of the mapped data is given as $G = \Phi^T\Phi = \left(G^{ij}\right)_{\substack{i=1,\ldots,C \\ j=1,\ldots,C}}$, where the submatrices $G^{ij} \in \Re^{K\times K}$ are defined as:

$$\begin{aligned} G^{ij} &= \left(k_{mn}^{ij}\right)_{\substack{m=1,\ldots,K \\ n=1,\ldots,K}} = <\phi\left(x_m^i\right), \phi\left(x_n^j\right)> \\ &= k\left(x_m^i, x_n^j\right)_{\substack{m=1,\ldots,K \\ n=1,\ldots,K}}. \end{aligned} \tag{15}$$

In the above equation $k(.,.)$ represents the kernel function, and one can easily create different decision boundaries around the query sample by simply using various distance metrics in the kernel function evaluations.

Our aim is to find the basis vectors for the intersection subspaces $N\left(S_i^\Phi\right) \cap R\left(S_T^\Phi\right)$, $i = 1, \ldots, C$, for each class. Then these basis vectors are employed for computation of the relative distances which are then used for labeling x_q. To find the basis vectors, we follow the previously mentioned steps; we first transform all nearest neighbors onto $R\left(S_T^\Phi\right)$ using Kernel PCA, then find the null spaces of class scatters in the transformed space.

The algorithm for NHKNN method can thus be summarized as follows:

Step 1: For each class, find the K-nearest neighbors to the query x_q using the selected distance metric.

Step 2: Transform all nearest neighbors onto $R\left(S_T^\Phi\right)$ using Kernel PCA. Let $\widetilde{G}$ be the kernel matrix of the centered mapped samples [20]. If we apply eigen-decomposition to $\widetilde{G}$, we obtain:

$$\begin{aligned} \widetilde{G} &= G - 1_M G - G 1_M + 1_M G 1_M \\ &= U \Lambda U^T \in \Re^{M\times M}, \end{aligned} \tag{16}$$

where $\Lambda \in \Re^{r\times r}$ is the diagonal matrix of nonzero eigenvalues ($r \le M-1$) and U is the matrix of normalized eigenvectors associated with Λ. The orthogonal matrix that projects centered neighbors onto $R\left(S_T^\Phi\right)$is $\Lambda^{-1/2}U^T(\Phi - \Phi 1_M)^T$.

Step 3: Compute the local scatter matrix of each class in the transformed space. The new scatter matrix $\widetilde{S}_i^\Phi \in \Re^{r\times r}$ for each class in the reduced space becomes:

$$\begin{aligned} \widetilde{S}_i^\Phi &= \left((\Phi - \Phi 1_M)U\Lambda^{-1/2}\right)^T \widetilde{S}_i^\Phi (\Phi - \Phi 1_M)U\Lambda^{-1/2} \\ &= \Lambda^{-1/2}U^T\widetilde{G}^{(i)}\widetilde{G}^{(i)T}U\Lambda^{-1/2} \quad i = 1, ..., C. \end{aligned} \tag{17}$$

Here, the matrix $\widetilde{G}^{(i)} \in \Re^{M\times K}$ is written as:

$$\begin{aligned} \widetilde{G}^{(i)} &= G^{(i)} - G^{(i)}1_K - 1_M G^{(i)} + 1_M G^{(i)} 1_K \\ &= \left(G^{(i)} - 1_M G^{(i)}\right)(I - 1_K), \end{aligned} \tag{18}$$

where the matrix $G^{(i)} \in \Re^{M\times K}$ is given by $G^{(i)} = \Phi^T\Phi^{(i)} = \left(G^{(i)j}\right)_{j=1,\ldots,C}$, and each submatrix $G^{(i)j} \in \Re^{K\times K}$ is defined as:

$$\begin{aligned} G^{(i)j} &= \left(k_{mn}^{(i)j}\right)_{\substack{m=1,..,K \\ n=1,\ldots,K}} = <\phi\left(x_m^j\right), \phi\left(x_n^i\right)> \\ &= k\left(x_m^j, x_n^i\right)_{\substack{m=1,\ldots,K \\ n=1,\ldots,K}}. \end{aligned} \tag{19}$$

Step 4: For each class, find an orthonormal basis for the null space of $\widetilde{S}_i^{\Phi}$. Let $Q^{(i)}$ be a matrix whose columns are the computed basis vectors such that:

$$Q^{(i)T}\widetilde{S}_i^{\Phi}Q^{(i)} = 0, \quad i = 1, ..., C \tag{20}$$

Step 5: The final matrix of basis vectors $W^{(i)}$, whose columns span the intersection subspace of the i-th class, is:

$$W^{(i)} = (\Phi - \Phi 1_M)U\Lambda^{-1/2}Q^{(i)}, \quad i = 1, \ldots, C. \tag{21}$$

The number of basis vectors spanning the intersection subspaces is determined by the dimensionality of $N\left(\widetilde{S}_i^{\Phi}\right)$ for each class. After performing the projection, all samples in $V_i^K(x_q)$ give rise to the local common vector of that class, given as:

$$\begin{aligned}\Omega_{\text{com}}^{(i)} &= W^{(i)T}\phi\left(x_m^i\right) \\ &= Q^{(i)T}\Lambda^{-1/2}U^T\widetilde{l}_m^i, i = 1, ..., C, \quad m = 1, ..., K,\end{aligned} \tag{22}$$

where $\widetilde{l}_m^i = \left(l_m^i - 1_M l_m^i\right) \in \Re^M$ and $l_m^i \in \Re^M$ is a vector with entries $k\left(x_n^j, x_m^i\right)_{\substack{j=1,\ldots,C \\ n=1,\ldots,K}}$. Note that the common vector given in Eq. 22 is independent of the sample index m, and hence one can choose any sample from $V_i^K(x_q)$ to obtain the corresponding local common vector. To recognize a given query sample, we compute the projection of the query sample by:

$$\begin{aligned}\Omega_{\text{query}}^{(i)} &= W^{(i)T}\phi\left(x_q\right) \\ &= Q^{(i)T}\Lambda^{-1/2}U^T\widetilde{l}_q\end{aligned}, \quad i = 1, ..., C, \tag{23}$$

where $\widetilde{l}_q = \left(l_q - 1_M l_q\right) \in \Re^M$ and $l_q \in \Re^M$ is a vector with entries $k\left(x_m^i, x_q\right)_{\substack{i=1,\ldots,C \\ m=1,\ldots,K}}$. Finally, we assign the query to the class with the closest local common vector.

The linearly constructed manifolds in the nonlinearly mapped higher-dimensional space correspond to nonlinear manifolds with complex structures in the original sample space. However, computing the distances between the query sample and these nonlinear manifolds is still straightforward in the sense described above, which makes the proposed technique very attractive.

In a similar manner, the CKNN method can be generalized to the nonlinear case by employing convex hulls rather than affine hulls in the Kernel PCA transformed space. The computations are as detailed in Section 2.2, but with an initial projection onto the Kernel PCA space. We call this method nonlinear CKNN (NCKNN) method.

3 The LDCV Method and its Nonlinear Counterpart

A variation of the HKNN method is obtained when the local difference subspace of each class is constructed using the combined linearly independent difference vectors of all nearest neighbors. This approach assumes that all classes have similar local variations since they are represented by the same subspace around each query point. As a result, linear decision boundaries are obtained around the query points in contrast to the quadratic decision boundaries of HKNN. This approach is similar to the method proposed in [3] in the sense that a linear discriminant analysis is used to deform the local metric based on linear manifolds. But, as in HKNN, this approach uses the null space of the involved covariance matrix, not the span. The method is also similar to Kim and Kittler's method [14], in which the global nonlinear structure of each face class is approximated by combining local linear manifolds. However, it should be noticed that the method proposed in [14] has separate training and test phases whereas our proposed scheme has no separate training phase—it is entirely instance-based by nature.

As the local difference subspace of each class is equal to the range of the scatter matrix S_i^K of samples coming from $V_i^K(x_q)$, the new combined difference subspace is equal to the range of the combined within-class scatter matrix $S_W^K = \sum_{i=1}^{C} S_i^K$ of the nearest neighbors in the vicinity of the query sample. As a consequence, the new indifference subspace is the null space of the within-class scatter matrix of selected neighbors. When these neighbors are projected onto the null space of the within-class scatter matrix, they give rise to unique common vectors representing the classes as in HKNN method. So, the decision function for a given query can be written as:

$$g\left(x_q\right) = \arg\min_{i=1,\ldots,C}\left(\left\|P_{\text{NS}}^W\left(x_q - \mu_i\right)\right\|\right), \tag{24}$$

where P_{NS}^W is the orthogonal projection operator onto the null space of the combined within-class scatter matrix. It should be noticed that P_{NS}^W is the projection matrix onto the intersection of the null spaces of local class scatter matrices, i.e., $P_{\text{NS}}^W = P_{\text{NS}}^{(1)} \cap P_{\text{NS}}^{(2)} \cap \ldots \cap P_{\text{NS}}^{(C)}$ [13].

This approach has been applied to the face recognition problem globally, where it is called the Discriminative Common Vector method [13]. Thus, we call the new local approach the Local Discriminative Common Vector (LDCV) method. The LDCV method can be extended to the nonlinear case using the kernel trick in the same way as NHKNN. Details of the kernelization of the global DCV method can be found in [21]. For the nonlinear LDCV (NLDCV) approach, one simply needs to use nearest neighbors instead of all available training data.

4 Experiments

In order to assess the performance of the methods, we test them on three data sets. We compare the proposed methods to NN, ADAMENN, HKNN, CKNN, SVM, and its local counterpart SVM-KNN. In all experiments, the one-against-all procedure is used to extend the two-class SVM to the multi-class classification. Cross-validation is used to estimate the kernel parameters and the number of nearest neighbors K unless a fixed validation set is not available. We search the range of values between 2 and 20 for K in the methods employing affine and convex hulls, and between 10 and 100 for SVM-KNN except for the Caltech101 database [23] where K is respectively set to 300 and 200 for the linear and nonlinear kernel functions.

4.1 Experiments on the United States Postal Service (USPS) Database

The USPS [24] database contains 9,298 16×16 gray-scale images of handwritten digits where 7,291 images are allocated for training+validation and the remaining 2007 for testing. Some samples are shown in Fig. 3. Since the training, validation, and test sets are fixed, the design parameters are set by using allocated validation set. After fixing parameters, the validation set is added to the training set, and the classification accuracies are assessed on the allocated test set.

Tangent distance has been shown to work well for the classification of hand-written digits [1, 25]. It improves the classification rate by compensating for small spatial affine transformations and changes in the thickness of the pen stroke. We use both the two-sided tangent distance and the Euclidean distance in our experiments as in [6]. We compute the tangent distances using C code downloaded from [26]. To incorporate tangent distances into the nonlinear approaches, we use a generalized Gaussian kernel $k(x,y) = \exp(-\mathrm{TD}(x,y)/q)$ where $\mathrm{TD}(x, y)$ denotes the two-sided tangent distance between two image vectors x and y. This kernel function does not satisfy the Mercer conditions, thus the kernel matrix is not necessarily positive semi-definite. There are several ways to handle this situation [27]—here we compute the most negative eigen-

Figure 3 Some samples from the USPS database.

Table 1 Classification rates of methods on the USPS database.

Methods	Classification rates (%)
NN	95.02
ADAMENN	97.36
HKNN, K=10	95.87
CKNN, K=20	96.02
LDCV, K=5	94.81
Linear SVM-KNN, K=10	95.97
Linear SVM	93.68
NN (TD)	96.97
Nonlinear SVM	NA
NHKNN, K=15, q=4e+5	97.56
NCKNN, K=15, q=4e+5	97.46
NLDCV, K=7, q=4e+5	97.07
SVM-KNN (TD), K=8, (Zhang et al. [6])	97.41

value and add its absolute value to the diagonal of the kernel matrix to make the kernel matrix positive semi-definite. The resulting classification rates are given in Table 1.

The best recognition rate of 97.56% is achieved by our NHKNN method using the tangent distances. Note that this is better than the human performance of 97.50% reported in [1]. Among the methods using Euclidean distances, CKNN gives the best results. ADAMENN also works well yielding 97.36% classification accuracy. Note that the classification rate of HKNN is different from the one reported in [4]. It is because we use direct distances between the query samples and the linear manifolds whereas [4] used a weight decay penalized solution for computing distances. All nonlinear approaches based on tangent distances show an improvement over their classical counterparts employing the Euclidean distances, which justifies the need for task specific distance metrics.

4.2 Experiments on the Image Segmentation Database

The UCI Image Segmentation Database [28] consists of samples drawn randomly from a database of seven outdoor images. The images are hand segmented to create a classification for every pixel. Each sample has a 3×3 region and 19 attributes. There are seven classes each having 330 samples. In our experiments, the attributes are normalized to lie in the interval [−1,1] and tenfold cross validation procedure is used to assess the generalization performance of the methods. We test both linear and nonlinear SVM classifiers. We use Gaussian kernels for all nonlinear approaches. The recognition rates and standard deviations are given in Table 2.

The best recognition rate is obtained by CKNN. Although NCKNN does not provide any improvement over

its linear counterpart, both nonlinear manifold based classifiers, NHKNN and NLDCV, outperform their linear counterparts. As discussed in Section 2, to apply HKNN method, the dimensionality of the sample space must be larger than the number of nearest neighbors (K) while for LDCV it must be larger than the number of total nearest neighbors (CK). Here, the dimensionality of the sample space is 19 and the number of classes is C=7. Therefore, we cannot use more than two neighbors for LDCV and 19 neighbors for HKNN. These methods typically perform best when the dimensionality is large compared to the number of data samples, i.e., $d >> K$ [13, 21]. So it is no surprise that LDCV performs worse than NN. On the other hand, this limitation does not apply to the nonlinear approaches since the prototype samples are mapped into a higher-dimensional feature space allowing the use of more nearest neighbors, which improves the recognition rates.

When the dimensionality of the sample space and the number of nearest neighbors are comparable, the affine or convex hulls of classes may overlap. If the query sample lies in these overlapping regions, the distance from the query to the corresponding classes is zero and the query cannot be labeled properly. Again, this problem is reduced in the proposed nonlinear approaches because the nearest neighbors are mapped into a higher-dimensional space. In particular, if the kernel ensures the strict positive definiteness of the kernel matrix, there is no overlap among the affine or convex hulls [21].

4.3 Experiments on the Caltech101 Database

The Caltech101 database [23] includes images of objects belonging to 101 visual categories. There are typically 40–800 images per category. The size of each image is roughly 300×200 pixels. Each class includes highly variable object poses under different lighting conditions and large intra-class variations. For our experiments we chose 40 classes from the database. The selected categories are shown in Fig. 4.

Table 2 Recognition rates of methods on the image segmentation database.

Methods	Recognition rates (%)	Standard deviations (σ)
NN	96.36	0.92
ADAMENN	95.85	0.87
HKNN, K=2	96.88	0.81
CKNN, K=15	97.31	1.02
LDCV, K=2	95.67	0.98
Linear SVM-KNN, K=75	96.49	0.95
Linear SVM	95.50	1.18
Nonlinear SVM, q=0.75	97.01	1.03
NHKNN, K=15, q=0.15	97.23	1.17
NCKNN, K=15, q=0.20	97.18	1.01
NLDCV, K=7, q=0.25	96.71	1.15
SVM-KNN, K=75, q=0.5	97.10	1.15

We represent the image samples using the "bag of features" representation. Introduced by Csurka et al. [29], such representations have been widely applied for both object classification and localization. In our approach, salient image patches are chosen at different positions and scales using a multi-scale Harris Laplace detector. Then, the chosen patches are described by using the scale invariant feature transform (SIFT) descriptors [30]. Following this process, all descriptors extracted from images are clustered by k-means clustering method and cluster means are considered as visual words forming the visual vocabulary. The size of the visual vocabulary is set to 2,000. To build image representation, each extracted descriptor is compared to the visual words and associated to the closest word. Based on these assignments, we build histograms, which are then used for classification of images.

In nonlinear approaches we use the generalized Gaussian kernel $k(x,y) = \exp(-\mathrm{CSD}(x,y)/q)$ based on chi-square (χ^2) distances between histograms where the χ^2 distance between two histograms $I_1 = (u_1, \ldots, u_d)$ and $I_2 = (v_1, \ldots, v_d)$ is $\mathrm{CSD}(I_1, I_2) = \frac{1}{2}\sum_{m=1}^{d}\left[(u_m - v_m)^2 \Big/ (u_m + v_m)\right]$. This function satisfies the Mercer's conditions [31], so we do not need to perturb the kernel matrix. The nearest neighbors around the query points are also found using χ^2 distance for all nonlinear local approaches.

To assess the recognition accuracy of the methods, we used the leave-one-out strategy since some object classes have few samples. The recognition rates are given in Table 3. Among all tested methods, the best recognition accuracy was achieved by global SVM classifiers. ADAMENN yielded the worst classification accuracy. Linear SVM-KNN method gave rise to inferior recognition accuracies compared to the linear manifold based approaches. All nonlinear local approaches outperformed their linear counterparts again showing the advantage of χ^2 distance, but the local approaches were not as good as the global SVM classifiers. It should be noted that there are around 50 samples with large intra-class variations for most of the selected categories and the input dimensionality is high (2,000). Thus the local neighborhoods are mostly empty. Consequently, local approaches yield poor recognition accuracies for small K, and the SVM classifier (which is equal to SVM-KNN when K is equal to the total number of samples in the training set) gave rise to the best recognition accuracies. ADAMENN's poor generalization performance can also be explained by the insufficient training samples. ADAMENN uses local posterior probability estimates to weigh each feature axis. Posterior probability estimates are not reliable in this case since the dimensionality is too high

Figure 4 The selected classes from the Caltech101 database.

and there are limited samples per class, which in turn degrades the classification performance.

4.4 Discussion

HKNN, CKNN, and LDCV methods differ from the local SVM-KNN in the way that their decision boundaries are constructed. HKNN locally fits a linear (affine) manifold to each class and uses quadratic distances to these manifolds. CKNN uses quadratic distances to the convex hulls of the training points in each class, and SVM-KNN learns a single linear separator that maximizes the gap between the classes. The decision boundaries of HKNN are quadratic while those of CKNN are piecewise linear and quadratic and contain the SVM-KNN boundary as some piecewise facet. In all linear and nonlinear manifold based approaches, the decision boundaries are constructed with respect to the query samples whereas they are built with completely ignoring query samples in SVM-KNN. This may cause unreliable assignments sometimes and the poor recognition rate of linear SVM-KNN on the Caltech database may be due to this phenomenon.

A closer look at the relation between recognition rates and ratio of the number of samples to the dimensionality of original sample space reveals interesting results and provides insights concerning suitable environments for the application of discussed local approaches. The assumption behind the local approaches is that class samples lie in some lower-dimensional smooth nonlinear manifold embedded in the original sample space, and this manifold can be approximated by locally linear structures such as affine or convex hulls. However this requires the manifold to be well sampled for modeling local structures correctly. As reported in [32], manifolds can be considered as well sampled if each sample in a class has on the order of $2l$ neighbors where l is the dimensionality of the underlying manifold of the class. The local approaches yielded good recognition rates for both the USPS and Image Segmentation data sets. Examining the ratio of the number of samples to the dimensionality of the original sample space shows that the above assumption is reasonable. On the other hand the local approaches are outperformed by the nonlinear SVMs for object categorization problem. Note that for most of the classes there are around 50 images with large intra-

Table 3 Recognition rates of methods on the Caltech101 database.

Methods	Recognition rates (%)
NN	31.90
ADAMENN	24.21
HKNN, K=20	66.78
CKNN, K=20	67.08
LDCV, K=15	57.28
Linear SVM-KNN, K=300	52.63
Linear SVM	70.23
NN (CSD)	56.95
Nonlinear SVM, q=1	70.23
NHKNN, K=20, q=0.20	69.05
NCKNN, K=20, q=0.20	68.06
NLDCV, K=15, q=0.25	67.07
SVM-KNN (CSD), K=200, q=1	67.69

class variance per class in the Caltech database, and this indicates that the class specific manifolds are not well sampled. Therefore, one has to check whether the underlying manifold is well sampled or not, before a possible application of local classifiers. There are various techniques to estimate the intrinsic dimensionality of manifolds [33–35], and these techniques could perhaps be used to test whether the local approaches would be suitable for a given classification task. In the light of our experimental study, it can be inferred that the proposed local approaches are good choices for recognizing hand-written digits.

All of the local approaches discussed in this paper share the advantages and disadvantages of prototype based classifiers: no prior training is required, which ideal for fast adaptation, multi-class problems are handled naturally. However, the classification is slow since a nearest neighbor search must run over the whole training set. For large training points, the computations required for finding distances to local affine/convex hulls are typically negligible compared to the cost of finding nearest neighbors. So the efficiency of the proposed methods depends on the training set size. In [4], the authors use a smaller but representative subset of the training data to speed up the HKNN algorithm. In particular, they employed support vectors obtained using an SVM classifier with a Gaussian kernel, and reported little loss of accuracy. Another idea for improving the efficiency is to create the local decision boundaries off-line during training. This scheme would compute the distance from each training sample to the affine or convex hulls of other rival classes. As we showed in the paper, computing the distances is straightforward by employing subspace concepts. Then, we choose local decision boundaries that maximize the average distance, thus combining local information with maximum margin concepts. We are currently working on this approach.

5 Conclusion

In this paper we first showed that the HKNN classifier can be formulated using subspaces. Then, based on the subspace formulation, the HKNN method has been extended to the nonlinear case using the kernel trick. However, the nonlinearization of the method is not trivial. The HKNN method needs to be modified before the nonlinearization. In addition, we introduced a variant of HKNN method, LDCV, which is better suited for classification problems with classes having similar variations. The LDCV method has also been extended to the nonlinear case using the same nonlinearization process. We tested the proposed nonlinear methods on three data sets. Experimental results demonstrate that the nonlinearization of the discussed local manifold based classifiers introduces improvements over their linear counterparts. Thus the proposed methods can find broad applications in classification areas where the Euclidean distances are not compatible.

Appendix

Theorem 1: Let P and $P_{\mathrm{NS}}^{(i)}$ be the projection matrices of the subspaces $R\left(S_T^K\right)$ and $N\left(S_i^K\right)$, $i=1,\ldots,C$, respectively. Then P and $P_{\mathrm{NS}}^{(i)}$ commute, i.e.:

$$P_{\mathrm{NS}}^{(i)}P = PP_{\mathrm{NS}}^{(i)}, \quad i=1,...,C.$$

Proof of the theorem is omitted since it can be derived as in the proof of Theorem 1 in [21].

Theorem 2: Assume that there are C classes in the training set. For a query x_q $\left\|P_{\mathrm{NS}}^{(i)}\left(x_q-\mu_i\right)\right\| \le \left\|P_{\mathrm{NS}}^{(j)}\left(x_q-\mu_j\right)\right\|$ implies that $\left\|P_{\mathrm{int}}^{(i)}\left(x_q-\mu_i\right)\right\| \le \left\|P_{\mathrm{int}}^{(j)}\left(x_q-\mu_j\right)\right\|$ for $i,j=1,\ldots,C$, and $i\neq j$.

Proof: We first recall several facts from [13] (see Lemma 1 of [13]). For each $i=1,\ldots,C$, it holds $N\left(S_T^K\right)\subset N\left(S_i^K\right)$, where $N(A)$ denotes the null space of a matrix A. Consequently, $N\left(S_T^K\right)$ and $R\left(S_i^K\right)$ are orthogonal, where $R\left(S_i^K\right)$ is the range of S_i^K. This implies the identity $(I-P)\left(I-P_{\mathrm{NS}}^{(i)}\right)=0$ or $(I-P)=(I-P)P_{\mathrm{NS}}^{(i)}$.

Thus, we can write:

$$\begin{aligned}\left\|P_{\mathrm{NS}}^{(i)}\left(x_q-\mu_i\right)\right\| &= \left\|PP_{\mathrm{NS}}^{(i)}\left(x_q-\mu_i\right)+(I-P)P_{\mathrm{NS}}^{(i)}\left(x_q-\mu_i\right)\right\| \\ &= \left\|PP_{\mathrm{NS}}^{(i)}\left(x_q-\mu_i\right)\right\|+\left\|(I-P)P_{\mathrm{NS}}^{(i)}\left(x_q-\mu_i\right)\right\| \\ &= \left\|PP_{\mathrm{NS}}^{(i)}\left(x_q-\mu_i\right)\right\|+\left\|(I-P)\left(x_q-\mu_i\right)\right\|.\end{aligned} \tag{25}$$

We now note that the vector $(I-P)\left(x_q-\mu_i\right)$ is the same for each class (i.e., it does not depend on the class index i) since we have shown in [21] that $(I-P)\mu_i$ is a so-called common vector for the class consisting of all samples in $V=\left\{x_m^i\right\}_{m=1,i=1}^{K,C}$ and that in fact $(I-P)x$ is the same vector for all x in the affine hull of V.

Thus, we have shown that:

$$\left\|P_{\mathrm{NS}}^{(i)}\left(x_q-\mu_i\right)\right\| = \left\|PP_{\mathrm{NS}}^{(i)}\left(x_q-\mu_i\right)\right\|+\|v\|, \tag{26}$$

for some vector v independent of the class index i. The assertion of Theorem 2 now immediately follows from this fact. □

References

1. Simard, P., Le Cun, Y., Denker, J., & Victorri, B. (1998). *Transformation invariance in pattern recognition—tangent distance and tangent propagation, lecture notes in computer science* (vol. 1524, pp. 239–274). Berlin: Springer.
2. Peng, J., Heisterkamp, D. R., & Dai, H. K. (2003). LDA/SVM Driven Nearest Neighbor Classification. *IEEE Trans Neural Netw*, *14*, 940–942. doi:10.1109/TNN.2003.813835.
3. Hastie, T., & Tibshirani, R. (1996). Discriminant adaptive nearest neighbor classification. *IEEE Trans. PAMI*, *18*(6), 607–616.
4. Vincent, P., & Bengio, Y. (2001). K-local hyperplane and convex distance nearest neighbor algorithms. *Adv Neural Inf Process Syst*, *14*, 985–992.
5. Domeniconi, C., & Gunopulos, D. (2002). *Efficient local flexible nearest neighbor classification*. In Proceedings of the 2nd SIAM International Conference on Data Mining.
6. Zhang, H., Berg, a. C., Maire, M., & Malik, J. (2006). *SVM-KNN: discriminative nearest neighbor classification for visual category recognition, in CVPR 2006* (pp. 2126–2136).
7. Peng, J., Heisterkamp, D. R., & Dai, H. K. (2004). Adaptive quasiconformal kernel nearest neighbor classification. *IEEE Trans Pattern Anal Mach Intell*, *28*, 656–661. doi:10.1109/TPAMI.2004.1273978.
8. Domeniconi, C., Peng, J., & Gunopulos, D. (2002). Locally adaptive metric nearest-neighbor classification. *IEEE Trans Pattern Anal Mach Intell*, *24*, 1281–1285. doi:10.1109/TPAMI. 2002.1033219.
9. Olkun, O. (2004). *Protein fold recognition with K-local hyperplane distance nearest neighbor algorithm*. In Proceedings of the 2nd European Workshop on data Mining and Text Mining in Bioinformatics, pp. 51–57.
10. Hinton, G. E., Dayan, P., & Revow, M. (1997). Modeling the manifolds of images of handwritten digits. *IEEE Trans Neural Netw*, *18*, 65–74. doi:10.1109/72.554192.
11. Roweis, S. T., & Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, *290*, 2323–2326. doi:10.1126/science.290.5500.2323.
12. Verbeek, J. (2006). Learning non-linear image manifolds by global alignment of local linear models. *IEEE Trans PAMI*, *28*, 1236–1250.
13. Cevikalp, H., Neamtu, M., & Wilkes, M. (2005). Discriminative common vectors for face recognition. *IEEE Trans PAMI*, *27*, 4–13.
14. Kim, T.-K., & Kittler, J. (2005). Locally linear discriminant analysis for multimodally distributed classes for face recognition with a single model image. *IEEE Trans PAMI*, *27*, 318–327.
15. Fitzgibbon, A. W., & Zisserman, A. (2003). *Joint manifold distance: a new approach to appearance based clustering*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
16. Zhang, J., Marszalek, M., Lazebnik, S., & Schmidt, C. (2006). *Local features and kernels for classification of texture and object categories: a comprehensive study*. In Proceedings of the Computer Vision and Pattern Recognition Workshop.
17. Tenenbaum, J. B., Silva, V., & Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, *290*, 2319–2323. doi:10.1126/science.290.5500.2319.
18. Gulmezoglu, M. B., Dzhafarov, V., & Barkana, A. (2001). The common vector approach and its relation to principal component analysis. *IEEE Trans Speech Audio Process*, *9*(6), 655–662. doi:10.1109/89.943343.
19. Boyd, S. (2004). *Convex optimization* pp. 399–401. Cambridge, UK: Cambridge University Press.
20. Schölkopf, B., Smola, A. J., & Muller, K. R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput*, *10*, 1299–1319. doi:10.1162/089976698300017467.
21. Cevikalp, H., Neamtu, M., & Wilkes, M. (2006). Discriminative common vector method with kernels. *IEEE Trans Neural Netw*, *17*, 1550–1565. doi:10.1109/TNN.2006.881485.
22. Xu, J., & Zikatanov, L. (2002). The method of alternating projections and the method of subspace corrections in hilbert space. *J Am Math Soc*, *15*, 573–597. doi:10.1090/S0894-0347-02-00398-3.
23. Fei-Fei, L. Fergus, R., & Perona, P. (2004) *Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories*. In Proceedings of the IEEE CVPR Workshop of Generative Model Based Vision.
24. USPS dataset of handwritten characters created by the US Postal Service. Retrieved from ftp://ftp.kyb.tuebingen.mpg.de/pub/bs/data.
25. Keysers, D., Dohmen, J., Theiner, T., & Ney, H. (2000). *Experiments with an extended tangent distance*. In Proceedings of the 15th International Conference on Pattern Recognition, vol. 2, pp. 38–42.
26. C codes for computing tangent distances. Retrieved from http://www-i6.informatik.rwth-aachen.de/~keysers/td/.
27. Golub, G. H., & Loan, C. F.-V. (1996). *Matrix computations* (3rd ed.). Baltimore, MD: Johns Hopkins University Press.
28. UCI—benchmark repository—a huge collection of artificial and real world data sets. University of California Irvine. Retrieved from http://www.ics.edu/~mlearn/MLRepository.html.
29. Csurka, G., Dance, C., Fan, L., Willamowski, J., Bray, C. (2004). *Visual categorization with bags of keypoints*. In Proceedings of the ECCV Workshop on Statistical Learning for Computer Vision.
30. Lazebnik, S., Schmid, C., & Ponce, J. (2005). A sparse texture representation using local affine regions. *IEEE Trans PAMI*, *27*(8), 1265–1278.
31. Fowlkes, C., Belogie, S., Chung, F., & Malik, J. (2004). Spectral grouping using the Nystrom method. *IEEE Trans PAMI*, *26*, 1–12.
32. Saul, L. K., & Roweis, S. T. (2003). Think globally, fit locally: unsupervised learning of low dimensional manifolds. *J Mach Learn Res*, *4*, 119–155.
33. Levina, E., & Bickel, P. J. (2005). Maximum likelihood estimation of intrinsic dimension. In L. K. Saul, Y. Weiss, & L. Bottou (Eds.), *Advances in neural information processing system*, *17* (pp. 777–784). Cambridge, MA: MIT Press.
34. Camastra, F., & Vinciarelli, A. (2002). Estimating the intrinsic dimension of data with a fractal-based method. *IEEE Trans Pattern Anal Mach Intell*, *24*(10), 1404–1407. doi:10.1109/TPAMI.2002.1039212.
35. Fukunaga, K., & Olsen, D. R. (1971). An algorithm for finding intrinsic dimensionality of data. *IEEE Trans Comput*, *C-20*, 176–183. doi:10.1109/T-C.1971.223208.

Hakan Cevikalp received the B.S. and M.S. degrees from the Electrical and Electronics Engineering Department of Eskisehir

Osmangazi University, Eskisehir, Turkey in 1999 and 2001, respectively. He received a Ph.D. degree in Electrical Engineering and Computer Science from Vanderbilt University in 2005. He worked as a post-doctoral researcher at Learning and Recognition in Vision (LEAR) team of Inria Rhone-Alpes, France and Rowan University, USA in 2007 and 2008, respectively. He is currently an assistant professor in Electrical and Electronics Engineering Department of Eskisehir Osmangazi University. His research interests include pattern recognition, neural networks, optimization, image and signal processing and computer vision.

Diane Larlus received a M.S. degree in image, vision and robotics from the National Polytechnic Institute of Grenoble (INPG, France) and prepared her Ph.D. in the LEAR group of the INRIA-Grenoble laboratory, in France. Since November 2008, she is a postdoctoral fellow at the Multimodal Interactive Systems group, in the Darmstadt University of Technology, in Germany. Her research focuses mainly on object recognition and segmentation, image classification and machine learning.

Marian Neamtu is a professor of mathematics at Vanderbilt University in Nashville, Tennessee. He received his M.Sc. in Mechanical Engineering in 1988 from the Slovak Technical University, Slovakia. He received a Ph.D. in mathematics in 1991 from the University of Twente, The Netherlands. His areas of expertise include numerical analysis, approximation theory, and statistics.

Bill Triggs is a CNRS researcher who works mainly on machine learning based approaches to understand images and other sensed data. He leads the AI (Apprentissage et Interfaces) team in the Laboratoire Jean Kuntzmann (LJK) in Grenoble, France, and he is also the deputy director of LJK, coordinator of the EU research project CLASS on unsupervised image and text understanding, and coordinator of the CNRS partner of the EU network of excellence PASCAL 2.

Frédéric Jurie is a full Professor at the University of Caen, France. He received his Ph.D. degrees in Computer Science from the University of Clermont-Fd. He joined the INRIA Lear group in 2003 and moved to the University of Caen in 2007. His researches are concerned with image understanding, especially image classification and object recognition. Other areas of past or current research include movement and object detection, and visual tracking. Dr. Jurie has served on variety of workshop and conference program committees, such as CVPR, ICCV or ECCV conferences. He is the author of over 80 publications in computer vision and related fields.