Local Subspace Classifiers: Linear and Nonlinear Approaches

Hakan Cevikalp, Diane Larlus, Matthijs Douze, Frédéric Jurie

▶ **To cite this version:**

Hakan Cevikalp, Diane Larlus, Matthijs Douze, Frédéric Jurie. Local Subspace Classifiers: Linear and Nonlinear Approaches. MLSP 2007 - IEEE Workshop on Machine Learning for Signal Processing, Aug 2007, Thessaloniki, Greece. pp.57-62, 10.1109/MLSP.2007.4414282 . hal-00203992

HAL Id: hal-00203992
https://hal.science/hal-00203992

Submitted on 21 Jan 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Local Subspace Classifiers: Linear and Nonlinear Approaches

Hakan Cevikalp, *Member, IEEE*, Diane Larlus, Matthijs Douze, and Frederic Jurie, *Member, IEEE*

***Abstract*—The K-local hyperplane distance nearest neighbor (HKNN) algorithm is a local classification method which builds nonlinear decision surfaces directly in the original sample space by using local linear manifolds. Although the HKNN method has been successfully applied in several classification tasks, it is not possible to employ distance metrics other than the Euclidean distances in this scheme, which can be considered as a major limitation of the method. In this paper we formulate the HKNN method in terms of subspaces. Advantages of the subspace formulation of the method are two-fold: First, it enables us to propose a variant of the HKNN algorithm, the Local Discriminative Common Vector (LDCV) method, which is more suitable for classification tasks where classes have similar intra-class variations. Second, the HKNN method along with the proposed method can be extended to the nonlinear case based on subspace concepts. As a result of the nonlinearization process, one may utilize a wide variety of distance functions in those local classifiers. We tested the proposed methods on several classification tasks. Experimental results show that the proposed methods yield comparable or better results than the Support Vector Machine (SVM) classifier and its local counterpart SVM-KNN.**

I. Introduction

ALTHOUGH being a very old and simple classification method, the Nearest Neighbor (NN) approach is among the most successful and robust methods for many classification problems. In this naïve approach, a query sample is assigned to the same class which includes the closest prototype sample. Various distance functions can be used to measure the closeness, such as the Euclidean or Mahalanobis distance. It was theoretically shown that for large number of samples in the training set, the NN rule exhibits good performance. In particular, the error committed by the NN rule is at most twice the Bayesian error. Additionally, it was empirically observed that the NN classifier with a well chosen distance metric outperforms more sophisticated classifiers in many situations [1-3].

Despite its advantages, the NN algorithm tends to yield a poor generalization ability with limited number of samples in high-dimensional spaces. *Hole artifacts* occur in the decision surface for such cases, which in turn introduces severe bias and reduces the generalization performance [4]. In order to overcome this pitfall, various methods have been proposed in the literature [4], [5], [6], [7]. Among these, the HKNN method was shown to work well in several classifications tasks [4], [8]. In this approach, each class is considered as a low-dimensional smooth manifold embedded in a high-dimensional space. In this setting, it is reasonable to assume that these manifolds are locally linear. In the case of limited training data, new points are fantasized to approximate each manifold locally. This process is accomplished using the nearest neighbors in the vicinity of a query sample to construct local linear manifolds for each class. Then the query sample is classified based on the distances from these local linear manifolds. In this way negative effects of the sparseness of the training data are reduced, and a significant improvement is obtained in recognition accuracies. Zhang *et al*. proposed a similar local classifier method called the SVM-KNN [5]. In this method they train an SVM classifier on the nearest neighbors using various distance functions. While utilizing various distance metrics may sound appealing, a further decrease of available prototypes through SVM is undesirable since the extracted support vectors may not model the decision boundaries correctly.

Recently, the development of specialized distance functions for some classification tasks has emerged as a fruitful line of research. Consequently, significant improvements were achieved by incorporating task specific distance metrics [5], [9]. For instance, the HKNN algorithm achieved the best recognition rate among all methods discussed in [5] for the Euclidean distances, but it was outperformed by the SVM-KNN method utilizing alternative distance metrics. Zhang *et al*. [9] reported that the Chi-Square and Earth Mover's distances show a significant improvement over linear kernels for classification of image histograms. All these results verify the hypothesis that some distance metrics are better suited for special classification tasks and one should exploit the advantages of using various distance functions in a classification task. However, the standard HKNN algorithm operates in the original sample space and it is not possible to utilize distance functions other than the Euclidean distances in this scheme.

In this paper we propose an elegant nonlinearization process which extends the capability of the HKNN method such that a wide variety of distances can be utilized in this scheme. As a consequence, the nearest neighbors can be 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 linearity assumption of the manifolds is more likely to be satisfied. The nonlinearization process also enables us to apply the HKNN approach to new classification tasks in which direct application is not

Manuscript received March 12, 2007. The authors are with the Learning and Recognition in Vision (LEAR) team of Inria Rhone-Alpes, France. (Emails: cevikalp, diane.larlus, douze, frederic.jurie@inrialpes.fr, Phone: +33 476615447; Fax: +33 476615454).

feasible. Although the constructed manifolds correspond to nonlinear structures in the original sample space, finding distances to these manifolds is still straightforward due to their linear nature in the mapped space. In this study, we also introduce a variation of the HKNN method, called the Local Discriminative Common Vector (LDCV) method, which is more suitable for classification tasks where classes have similar variations. Then, it is extended to the nonlinear case using the same nonlinearization process.

The remainder of the paper is organized as follows: In Section 2, we formulate the HKNN method in terms of subspaces and generalize it to the nonlinear case using the kernel trick. In Section 3, the LDCV method is introduced and kernelized. Section 4 describes the data sets and experimental results. Finally, we draw our conclusion in Section 5.

II. KERNELIZATION OF THE HKNN METHOD

We can map the nearest neighbor samples into a higher-dimensional space $\Im$ to extend the standard HKNN method to the nonlinear case. A major advantage of the nonlinearization process is that it allows using distance metrics other than the Euclidean distances during transformation. The Euclidean distances may not be compatible for some classification tasks as demonstrated in the experiments, and the nonlinearization process offers the capability of incorporating different distance functions in such cases. In order to incorporate different distance metrics in the HKNN method, we first formulate the method in terms of subspaces. Then, the method is extended to the nonlinear case using subspace concepts and the kernel trick.

A. Formulation of the HKNN Method in Terms of Subspaces

In the HKNN method, the first step is to find the closest *K* neighbors to a query sample for each class. Then these neighbors are used to construct the local linear manifolds of the classes. Finally the query sample is assigned to the class associated with the closest manifold. Next, we will show that the HKNN method can be seen as a local subspace classifier.

Suppose there are *C* classes in the training set. Let $V_i^K(x_q)=\{x_1^i, x_2^i,..., x_K^i\}$ denotes the set of *K* nearest samples of the query sample $x_q \in \Re^d$ in the training set, belonging to the *i*-th class. Here we consider the dimensionality of the sample space *d* is larger than or equal to *K*, i.e., $d \geq K$. The local affine hull of each class is defined in terms of the closest *K* sample vectors as

$$LH_i^K(x_q)=\left\{p \mid p=\sum_{m=1}^{K}\alpha_m^i x_m^i, \alpha_m^i \in \Re, \sum_{m=1}^{K}\alpha_m^i=1\right\}, i=1,..,C. \quad (1)$$

Note that the above equation constructs the lowest-dimensional manifold which passes through all points of $V_i^K(x_q)$. In order to get rid of the constraint $\sum_{m=1}^{K}\alpha_m^i=1$ we can choose any reference point from $V_i^K(x_q)$, e.g., mean $\mu_i=(1/K)\sum_{m=1}^{K}x_m^i$, and rewrite the equation (1) as

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

where z_m^i are the linearly independent vectors obtained from the difference vectors $\{x_1^i-\mu_i, x_2^i-\mu_i,..., x_K^i-\mu_i\}$. Here l_i represents the number of linearly independent difference vectors and $l_i \leq K-1$. Note that there is no constraint on the sum of new coefficients β_m^i. The linearly independent difference vectors $z_1^i, z_2^i,..., z_{l_i}^i$ span the *difference subspace* of the local vector set $V_i^K(x_q)$ [10]. It can be shown that the difference space and the range space of the covariance matrix of samples in $V_i^K(x_q)$ are equivalent spaces [11].

In order to classify a query point x_q, the minimum distances between the query vector and the local manifolds must be computed. Then the query sample is assigned to the class whose manifold is the closest to x_q. The minimum distance between x_q and each manifold is computed by,

$$\begin{aligned} d(x_q, LH_i^K(x_q)) &= \min_{p \in LH_i^K(x)} \| x_q - p \| \\ &= \min_{\beta \in \Re^{l_i}} \| x_q - \mu_i - Z^{(i)}\beta^{(i)} \| \end{aligned}, \; i=1,...,C, \quad (3)$$

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

$$\beta^{(i)} = inv(Z^{(i)^T}Z^{(i)})Z^{(i)^T}(x_q-\mu_i). \quad (4)$$

It should be noted that the matrix $Z^{(i)}inv(Z^{(i)^T}Z^{(i)})Z^{(i)^T}$ defines an orthogonal projection operator and in our case it is the orthogonal projection operator of the difference subspace of $V_i^K(x)$. Thus, we can rewrite (3) as

$$\begin{aligned} d(x_q, LH_i^K(x_q)) &= \| x_q - \mu_i - P^{(i)}(x_q-\mu_i) \| \\ &= \| (I-P^{(i)})(x_q-\mu_i) \|, \end{aligned} \quad (5)$$

where *I* is the identity matrix and $P^{(i)}$ is the orthogonal projection operator of the difference subspace of the *i*-th class. The matrix $P_{NS}^{(i)}=(I-P^{(i)})$ is called the orthogonal projection operator of the *indifference subspace* (the null space of the covariance matrix) of $V_i^K(x)$ [10]. Notice that the difference and indifference subspaces are orthogonal complements of each other. The projections of all samples and their affine combinations in $V_i^K(x)$ onto their corresponding indifference subspace produce a unique vector x_{com}^i representing that vector set. More formally,

$$x_{com}^i = P_{NS}^{(i)}x_{aff}^i, \; i=1,..,C, \quad (6)$$

where $x_{aff}^i \in LH_i^K(x_q)$. Thus the minimum distance of the test vector from each manifold can be written as

$$d(x_q, LH_i^K(x_q)) = \| P_{NS}^{(i)} x_q - x_{com}^i \|, \; i = 1,..,C. \quad (7)$$

The above formula shows that the minimum distance from a query sample to each local manifold constructed using the *K*-nearest neighbors is equal to the Euclidean distances between the local common vector of each class and the projection of x_q onto the local indifference subspace. 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 neighbors in this setting. In that sense the distance from the query sample to each class manifold is similar to the Mahalanobis distance and it may be considered as a variant of the Mahalanobis distance for rank deficient covariance matrices. As a result, the decision function for a given query sample x_q can be written as

$$g(x_q) = \arg\min_{i=1,...,C} (\| P_{NS}^{(i)} x_q - x_{com}^i \|). \quad (8)$$

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

B. Kernelization Process

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

$$S_i^K = \sum_{m=1}^{K} (x_m^i - \mu_i)(x_m^i - \mu_i)^T, \quad i = 1,...,C. \quad (9)$$

Note that the HKNN algorithm utilizes the null spaces of these local scatter matrices for classification of the query samples. Similarly, the local total scatter matrix S_T^K of all neighbors in the vicinity of the query sample is defined as

$$S_T^K = \sum_{i=1}^{C} \sum_{m=1}^{K} (x_m^i - \mu)(x_m^i - \mu)^T, \quad (10)$$

where μ is the mean of all neighbors.

We use the kernel trick in order to map the data into higher-dimensional space as in the Kernel PCA [12] approach. However the kernelization of the HKNN algorithm is not trivial since the method utilizes the null spaces of the covariance matrices of the mapped samples rather than the range spaces. In this case the HKNN algorithm cannot be formulated in terms of the dot products of the mapped samples as in the Kernel PCA approach. Therefore we will modify the original HKNN algorithm such that it can be formulated in terms of the dot products of the mapped samples in $\Im$. To this end, it should be noticed that the null space $N(S_T^K)$ of the local total scatter matrix of all nearest neighbors does not contain any discriminative information for classification. This is because the projections of all neighbors onto this subspace give rise to the same vector [13]. Therefore, without loss of generality, this subspace can be discarded from our consideration in the HKNN method. Then, the new local subspace representing each class around the vicinity of the query sample can be defined as the intersection of the local null space of that class' scatter matrix and the range space of the local total scatter matrix, i.e., $N(S_i^K) \cap R(S_T^K)$, $i = 1,...,C$.

In order to use the intersection subspaces for classification of query samples, we have to compute the projection matrices $P_{\text{int}}^{(i)}$, $i = 1,...,C$, of those intersections. The projection matrix $P_{\text{int}}^{(i)}$ of the intersection $N(S_i^K) \cap R(S_T^K)$ for each class can be found as

$$P_{\text{int}}^{(i)} = P_{NS}^{(i)} P = P P_{NS}^{(i)}, \quad i = 1,...,C, \quad (11)$$

since the projection matrices of $N(S_i^K)$ and $R(S_T^K)$ commute. Here P represents the projection matrix of $R(S_T^K)$. Notice that, in general, the projection matrix of any intersection can not be obtained using (11) if the projection matrices of the associated subspaces do not commute. As a result, we can first project all nearest neighbors onto $R(S_T^K)$ and find the null spaces of the projected samples in the transformed space so as to compute the basis vectors spanning the local intersection subspaces. In order to extend the HKNN algorithm to the nonlinear case we are going to apply this procedure in the mapped space as described below.

C. Nonlinear HKNN (NHKNN) Method

This method consists in mapping the nearest neighbors around the query sample into an implicit higher-dimensional space $\Im$ using a nonlinear kernel mapping function and then applying the above procedure in the mapped space. Using intersection subspaces in the mapped space allows us to formulate the method in terms of the dot products of the mapped samples. Kernel functions are used to compute those dot products as in the other methods using the kernel trick. As a result, the mapping function and the mapped samples are not used explicitly, which makes the method computationally feasible.

Let $\Phi = [\Phi^{(1)}, \Phi^{(2)}, \ldots, \Phi^{(C)}]$ represent the matrix whose columns are the mapped nearest neighbors in $\Im$ where $\Phi^{(i)} = [\phi(x_1^i), \phi(x_2^i), ..., \phi(x_K^i)]$ is the matrix whose columns are mapped neighbor samples of the *i*-th class. Suppose $M = CK$ is the total number of 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 samples in $\Im$ are given by

$$\begin{aligned} S_i^\Phi &= \sum_{m=1}^{K} (\phi(x_m^i) - \mu_i^\Phi)(\phi(x_m^i) - \mu_i^\Phi)^T \\ &= (\Phi^{(i)} - \Phi^{(i)} 1_K)(\Phi^{(i)} - \Phi^{(i)} 1_K)^T, \quad i = 1,...,C, \end{aligned} \quad (12)$$

$$\begin{aligned} S_T^\Phi &= \sum_{i=1}^{C} \sum_{m=1}^{K} (\phi(x_m^i) - \mu^\Phi)(\phi(x_m^i) - \mu^\Phi)^T \\ &= (\Phi - \Phi 1_M)(\Phi - \Phi 1_M)^T, \end{aligned} \quad (13)$$

where μ_i^Φ is the mean of mapped nearest neighbor samples in the *i*-th class, μ^Φ is the mean of all mapped 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 $G = \Phi^T\Phi = (G^{ij})_{\substack{i=1,...,C\\ j=1,...,C}}$, where each submatrix $G^{ij} \in \Re^{K\times K}$ is defined as

$$G^{ij} = (k^{ij}_{mn})_{\substack{m=1,...,K\\ n=1,...,K}} =< \phi(x^i_m), \phi(x^j_n) >= k(x^i_m, x^j_n)_{\substack{m=1,...,K\\ n=1,...,K}}. \quad (14)$$

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

Our aim is to find the basis vectors for the intersection subspaces $N(S_i^\Phi) \cap R(S_T^\Phi)$, $i = 1,...,C$, for each class. To find these basis vectors, we follow the previously mentioned steps; we first transform all nearest neighbor samples onto $R(S_T^\Phi)$ and then find the null spaces of classes in the transformed space. The transformation of nearest neighbors onto $R(S_T^\Phi)$ can be done easily by employing the Kernel PCA method. Then we find the vectors spanning the null spaces of the scatter matrices of the transformed samples.

The algorithm for the nonlinear HKNN method can be summarized as follows:

Step 1: For each class, find the *K* closest samples to the query sample x_q.

Step 2: Transform all nearest neighbors onto $R(S_T^\Phi)$ using the Kernel PCA. Let $\widetilde{G}$ be the kernel matrix of the centered mapped samples [12]. If we apply eigen-decomposition to $\widetilde{G}$, we obtain

$$\widetilde{G} = G - 1_M G - G1_M + 1_M G1_M = U\Lambda U^T \in \Re^{M\times M}, \quad (15)$$

where Λ is the diagonal matrix of nonzero eigenvalues and *U* is the matrix of normalized eigenvectors associated to Λ. The matrix that transforms the samples onto $R(S_T^\Phi)$ is $(\Phi - \Phi 1_M)U\Lambda^{-1/2}$.

Step 3: Compute the local scatter matrix of each class in the transformed space. The new scatter matrix $\widetilde{S}_i^\Phi \in \Re^{r\times r}$ (*r* is the rank of $R(S_T^\Phi)$ and that cannot be larger than *M*-1) of each class in the reduced space becomes

$$\begin{aligned}\widetilde{S}_i^\Phi &= ((\Phi - \Phi 1_M)U\Lambda^{-1/2})^T S_i^\Phi (\Phi - \Phi 1_M)U\Lambda^{-1/2}\\ &= \Lambda^{-1/2}U^T\widetilde{G}^{(i)}\widetilde{G}^{(i)T}U\Lambda^{-1/2}, \quad i = 1,...,C.\end{aligned} \quad (16)$$

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

$$\begin{aligned}\widetilde{G}^{(i)} &= G^{(i)} - G^{(i)}1_K - 1_M G^{(i)} + 1_M G^{(i)}1_K\\ &= (G^{(i)} - 1_M G^{(i)})(I - 1_K),\end{aligned} \quad (17)$$

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

$$G^{(i)j} = (k^{(i)j}_{mn})_{\substack{m=1,...,K\\ n=1,...,K}} =< \phi(x^j_m), \phi(x^i_n) >= k(x^j_m, x^i_n)_{\substack{m=1,...,K\\ n=1,...,K}}. \quad (18)$$

Step 4: For each class, find a basis of the null space of $\widetilde{S}_i^\Phi$. This can be done by an eigen-decomposition. The normalized eigenvectors corresponding to the zero eigenvalues of $\widetilde{S}_i^\Phi$ form an orthonormal basis for the null space of $\widetilde{S}_i^\Phi$. Let $Q^{(i)}$ be a matrix whose columns are the computed eigenvectors corresponding to the zero eigenvalues, such that

$$Q^{(i)T}\widetilde{S}_i^\Phi Q^{(i)} = 0, \quad i = 1,...,C. \quad (19)$$

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

$$W^{(i)} = (\Phi - \Phi 1_M)U\Lambda^{-1/2}Q^{(i)}, \quad i = 1,...,C. \quad (20)$$

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

$$\begin{aligned}\Omega^{(i)}_{com} &= W^{(i)T}\phi(x^i_m)\\ &= Q^{(i)T}\Lambda^{-1/2}U^T\widetilde{l}^i_m,\ i = 1,...,C, m = 1,...,K,\end{aligned} \quad (21)$$

where $\widetilde{l}^i_m = (l^i_m - 1_M l^i_m) \in \Re^M$ and $l^i_m \in \Re^M$ is a vector with entries $k(x^j_n, x^i_m)_{\substack{j=1,...,C\\ n=1,...,K}}$. Note that the common vector given in (21) 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 feature vector of the query sample by

$$\begin{aligned}\Omega^{(i)}_{query} &= W^{(i)T}\phi(x_q)\\ &= Q^{(i)T}\Lambda^{-1/2}U^T\widetilde{l}_q, \quad i = 1,...,C,\end{aligned} \quad (22)$$

where $\widetilde{l}_q = (l_q - 1_M l_q) \in \Re^M$ and $l_q \in \Re^M$ is a vector with entries $k(x^i_m, x_q)_{\substack{i=1,...,C\\ m=1,...,K}}$. Then, we compare the Euclidean distances between the common vectors and the feature vector of the query sample for each class using (8), and the query sample is assigned to the class that minimizes this distance.

III. Local DCV Method and its Kernel Counterpart

We get a variation of the HKNN when the local difference subspace of each class is constructed using pooled linearly independent difference vectors of all classes. This approach assumes 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 HKNN algorithm in which the quadratic decision boundaries are obtained.

It is known that the local difference subspace of each class is equal to the range of the scatter matrix S_i^K of samples coming from $V_i^K(x_q)$. Therefore the new pooled difference subspace is equal to the range of the 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 the neighbors. When the nearest neighbors are projected onto the null space of the within-class scatter matrix, they give rise to unique common vectors representing classes as in the HKNN method. In that case, the decision function for a given query sample is written as

$$g(x_q) = \arg\min_{i=1,\ldots,C}(\| P_{NS}(x_q - \mu_i) \|), \quad (23)$$

where P_{NS} is the orthogonal projection operator of the null space of the within-class scatter matrix.

This approach has been applied to the face recognition problems globally and it has been called as the Discriminative Common Vector method [11]. Therefore, we will call the new local approach as the local Discriminative Common Vector (LDCV) method. The keen reader can refer to [11] for a detailed explanation on the DCV method.

The LDCV method can also be extended to the nonlinear case using the kernel trick similar to the NHKNN. Details of the kernelization of the global DCV method can be found in [13]. Note that the kernelization was done for global DCV in that paper, thus one should use the closest prototype samples in the nonlinear LDCV (NLDCV) approach instead of using all available training data.

IV. Experiments

In order to assess the performance of the proposed local methods, we tested them on 2 data sets. We compared the proposed methods to the NN, SVM and its local counterpart SVM-KNN. In all experiments, the one-against-all procedure has been used to extend classic two-class SVM problem to the multi-class recognition problem. To find the best parameters (kernel function parameters and K) of algorithms we followed the procedure described in [13].

A. Experiments on the USPS Database

The USPS database contains 9298 gray-scale images of handwritten digits where 7291 images are allocated for training and the remaining 2007 are allocated for testing. The size of each image is 16x16 and the human error rate is reported as 2.5% on this database [1]. Some samples from the USPS database are shown in Fig. 1. We employed the Euclidean and tangent distances in our experiments as in [5]. In order to incorporate the tangent distances in the nonlinear approaches, we used the generalized Gaussian kernel $k(x,y) = \exp(-TD(x,y)/q)$ where $TD(x,y)$ denotes the two-sided tangent distance between two image vectors x and y. The employed kernel function does not satisfy the Mercer conditions, thus the kernel matrix is not necessarily a positive semi-definite matrix. There are different ways to handle this situation. In our case we computed the most negative eigenvalue and added its absolute value to the diagonal of the kernel matrix in order to make the kernel matrix positive semi-definite. The computed error rates are given in Table I.

As can be seen in the table, the best recognition rate is achieved with our proposed method NHKNN. It is interesting that it is even better than the human recognition performance reported in [1]. All nonlinear approaches employing tangent distances show an improvement over their classical counterparts employing the Euclidean distances, which justify our initial claims on employing task specific distance metrics.

Fig. 1. Some samples from the USPS database.

TABLE I
Error Rates on the Usps Database

Methods	Error Rates (%)
NN	4.98
NN (TD)	3.03
HKNN, $K=10$	4.13
LDCV, $K=2$	5.19
Linear SVM	6.32
Nonlinear SVM	NA
NHKNN, $K=15$, $q=4e+5$	**2.44**
NLDCV, $K=7$, $q=4e+5$	2.93
SVM-KNN, $K=8$, (Zhang *et al*. [5])	2.59

B. Experiments on the Image Segmentation Database

The Image Segmentation Database [14] consists of samples randomly drawn from a database of seven outdoor images. The images are hand segmented to create a classification for every pixel. Each sample has a 3x3 region and 19 attributes. There are a total of 7 classes each having 330 samples. In our experiments, the attributes were normalized to lie in interval [-1,1] and 10-fold cross validation procedure has been used to assess the generalization performance of the methods. We tested both linear and nonlinear SVM classifiers. The Gaussian kernel has been employed in the proposed methods as well as in the nonlinear SVM and SVM-KNN. The recognition rates and standard deviations are given in Table II.

TABLE II
Recognition Rates on the Image Segmentation Database

Methods	Recognition Rates (%)
NN	96.36, $\sigma=0.92$
HKNN, $K=2$	96.88, $\sigma=0.81$
LDCV, $K=2$	95.67, $\sigma=0.98$
Linear SVM	95.50, $\sigma=1.18$
Nonlinear SVM, $q=0.75$	97.01, $\sigma=1.03$
NHKNN, $K=15$, $q=0.15$	**97.23**, $\sigma=1.17$
NLDCV, $K=7$, $q=0.25$	96.71, $\sigma=1.15$
SVM-KNN, $K=75$, $q=0.5$	97.10, $\sigma=1.15$

In terms of the recognition accuracy, the best recognition rate is obtained by the proposed nonlinear method NHKNN. Both nonlinear subspace classifiers outperform their linear counterparts. As stated in Section 2, the dimensionality of the sample space must be larger than the number of nearest neighbors (K) to apply the HKNN method. Similarly, the dimensionality of the sample space must be larger than the total number of nearest neighbors (CK) for the LDCV

method. It was reported that the subspace methods perform best when the dimensionality is large compared to the number of data samples. Notice that the dimensionality of the sample space is 19 for the Image Segmentation database. Therefore, we could not employ many nearest neighbors in the local linear subspace approaches. As a result, the LDCV method performed worse than the NN method. On the other hand, this limitation does not exist in the nonlinear approaches since the nearest neighbors are mapped into a higher-dimensional feature space. Consequently, we employed more nearest neighbors in the nonlinear approaches, which improved the recognition rates.

C. Discussion and Future Work

The proposed classifiers share the same advantages of other prototype based classifiers (no training required, ideal for fast adaptation, natural handling of the multi-class case) However, the testing time is very slow as in those methods since the query sample must be compared to all available data samples to find the closest neighbors. Other computations for obtaining the subspace parameters of the nearest neighbors is negligible compared to finding the closest neighbors. Therefore the real-time efficiencies of the proposed methods depend on the training set size. In [4], the authors used 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 the Gaussian kernel. They reported similar recognition accuracy obtained using all data.

A further improvement can be achieved by a similar procedure. In this scheme, we first train an SVM classifier and extract representative support vectors. Then, we treat each support vector as a query point and compute the local subspace parameters using the nearest neighbors around the support vectors. Note that all these computations are performed offline. In real-time classification of a test sample, we find the closest support vector and use the associated subspace parameters for classification of the test sample. More than one support vector can be employed in this approach. Since the test sample is compared to only support vectors and all subspace parameters are pre-computed in this approach, a significant improvement on the real-time efficiency can be obtained. However, the effects of this procedure on the recognition performance must be checked before a possible application. We are currently working on this approach.

V. 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 was extended to the nonlinear case using the kernel trick. However, the nonlinearization of the method was not trivial. The HKNN method needed to be modified before the nonlinearization. In addition we introduced a variant of the HKNN method, which is called as the LDCV method. Then, the LDCV method was also extended to the nonlinear case using the same nonlinearization process. We tested the proposed nonlinear methods on two data sets. Experimental results demonstrate that the nonlinearization of the discussed subspace classifiers results in novel elegant methods, which can find broad applications in classification areas where the Euclidean distances are not compatible.

References

[1] P. Simard, Y. LeCun, and J. S. Denker, "Efficient pattern recognition using a new transformation distance," *in NIPS*, pp. 50-58, 1993.

[2] J. Peng, D. R. Heisterkamp, and H. K. Dai, "LDA/SVM driven neartest neighbor classification," *IEEE Trans. on Neural Networks*, vol. 14, pp. 940-942, 2003.

[3] T. Hastie and R. Tibshirani, "Discriminant adaptive nearest neighbor classification," *IEEE Trans. on PAMI*, vol. 18, no. 6, pp. 607-616, June 1996.

[4] P. Vincent and Y. Bengio, "K-local hyperplane and convex distance nearest neighbor algorithms," *in NIPS*, pp. 985-992, 2001.

[5] H. Zhang, a. C. Berg, M. Maire, and J. Malik, "SVM-KNN: discriminative nearest neighbor classification for visual category recognition," *in CVPR 2006*, pp. 2126-2136, 2006.

[6] C. Domeniconi and D. Gunopulos, "Efficient local flexible nearest neighbor classification," *in Proc. of the 2nd SIAM International Conference on Data Mining*, 2002.

[7] K. Q. Weinberger, J. Blitzer, and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *Advances in Neural Information Processing Systems 18*, MIT Press: Cambridge, MA, 2006.

[8] O. Olkun, "Protein fold recognition with K-local hyperplane distance nearest neighbor algorithm," *in proc. of the 2nd European Workshop on data Mining and Text Mining in Bioinformatics*, pp. 51-57, 2004.

[9] J. Zhang, M.. Marszalek, S. Lazebnik, and C. Schmid, "Local features and kernels for classification of texture and object categories: a comprehensive study," in Computer Vision and Pattern Recognition Workshop, 2006.

[10] M. B. Gulmezoglu, V. Dzhafarov, and A. Barkana, "The common vector approach and its relation to principal component analysis," *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 6, September 2001.

[11] H. Cevikalp, M. Neamtu, M. Wilkes, and A. Barkana, "Discriminative common vectors for face recognition," *IEEE Trans. on PAMI*,vol. 27, 4-13, January 2005.

[12] B. Schölkopf, A. J. Smola, and K. R. Muller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural computation*, vol. 10, 1299-1319, 1998.

[13] H. Cevikalp, M. Neamtu, and M. Wilkes, "Discriminative common vector method with kernels," *IEEE Transactions on Neural Networks,* vol. 17, pp. 1550-1565, 2006.

[14] "UCI-Benchmark repository-A huge collection of artificial and real world data sets," University of California Irvine, http://www.ics.edu/ ~mlearn/MLReposiyory.html.