

On Utilizing Search Methods to Select Subspace Dimensions for Kernel-Based Nonlinear Subspace Classifiers

Sang-Woon Kim, *Senior Member, IEEE*, and
B. John Oommen, *Fellow, IEEE*

Abstract—In Kernel-based Nonlinear Subspace (KNS) methods, the subspace dimensions have a strong influence on the performance of the subspace classifier. In order to get a high classification accuracy, a large dimension is generally required. However, if the chosen subspace dimension is too large, it leads to a low performance due to the overlapping of the resultant subspaces and, if it is too small, it increases the classification error due to the poor resulting approximation. The most common approach is of an ad hoc nature, which selects the dimensions based on the so-called cumulative proportion [13] computed from the kernel matrix for each class. In this paper, we propose a new method of systematically and efficiently selecting optimal or near-optimal subspace dimensions for KNS classifiers using a search strategy and a heuristic function termed the *Overlapping* criterion. The rationale for this function has been motivated in the body of the paper. The task of selecting optimal subspace dimensions is reduced to finding the best ones from a given problem-domain solution space using this criterion as a heuristic function. Thus, the search space can be pruned to very efficiently find the best solution. Our experimental results demonstrate that the proposed mechanism selects the dimensions efficiently without sacrificing the classification accuracy.

Index Terms—Kernel principal component analysis (kPCA), kernel-based nonlinear subspace (KNS) classifier, subspace dimension selections, state-space search algorithms.

1 INTRODUCTION

WE consider a fundamental (difficult) problem encountered in Kernel-based Nonlinear Subspace (KNS) methods, namely, that of determining the subspace dimensions of the classifier. Unfortunately, formal algorithmic methods to solve this problem are not yet known. In this paper, we shall reduce the problem to one of searching in an appropriately modeled solution space. The best dimension is thus obtained by searching within this solution space using the well-acclaimed search techniques found in Artificial Intelligence (AI) literature. However, rather than search the entire solution space, we advocate that, by using a *well-designed heuristic function*, this search process can be enhanced by intelligently pruning the search tree. All of these phases are novel to the field of selecting the KNS subspace dimensions and, in particular, the possibility of using heuristic-based search is unreported in this application.

To pose the problem in the right perspective, a brief overview of the general problem domain follows. The subspace method of pattern recognition is a technique in which the pattern classes are not primarily defined as bounded regions or zones in a feature space, but, rather, given in terms of linear subspaces defined by the basis vectors, one for each class,¹ ω_i [13]. The length of a vector projected onto the subspace associated with a class is

1. Where there is no confusion, we shall refer to the class ω_i by its index, i .

- S.-W. Kim is with the Department of Computer Science and Engineering, Myongji University, Yongin, 449-728 Korea. E-mail: kimswo@mju.ac.kr.
- B.J. Oommen is with the School of Computer Science, Carleton University, Ottawa, Canada K1S 5B6. E-mail: oommen@scs.carleton.ca.

Manuscript received 25 Nov. 2003; revised 23 June 2004; accepted 4 Aug. 2004.

Recommended for acceptance by E. Hancock.

For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0381-1103.

measured by the Principal Components Analysis (PCA). This involves computing the covariance (or scatter) matrix, its normalized eigenvectors, extracting their principal components, and, finally, computing the projections of a test pattern vector onto (or from) the class subspaces spanned by the subset of principal eigenvectors. Various subspace classifiers, such as CLAFIC (CLAss Feature Compression) [13], the learning subspace method [13], and the mutual subspace method [10], have been proposed in the literature. However, since the PCA is a linear algorithm, it essentially limits the use of subspace classifiers. To overcome this limitation, a kernel Principal Components Analysis (kPCA) was proposed in [11], [17]. The kPCA provides an elegant way of dealing with nonlinear problems in an input space R^N by mapping them to linear ones in a feature space, F . That is, a dot product in space R^N corresponds to mapping the data into a possibly high-dimensional product space F by a nonlinear map $\Phi: R^N \rightarrow F$ and taking the dot product in this space.

Recently, using the kPCA, kernel-based subspace methods have been proposed, such as the Subspace method in Hilbert Space (SHS) [20], the Kernel-based Nonlinear Subspace (KNS) method [9], and the Kernel-Based Learning (KBL) algorithms [11] (among others). All of them utilize the *kernel trick* to obtain the kernel PCA components by solving a similar linear eigenvalue problem as explained above for the linear PCA. The only difference is that the size of the problem is decided by the *number* of data points and not by the dimension. The above studies report that the performance of nonlinear subspace methods is better than that of linear methods possessing the same number of principal components. In kPCA, to map the data set $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ (where each $\mathbf{x}_i \in R^N$) into a feature space F , we have to work with an $n \times n$ matrix, K , the so-called kernel matrix. Observe that the dimension of K is equivalent to the number of data points, n .

To solve the computational problem, a number of methods have been proposed, such as the techniques proposed by Achlioptas et al. [1], [2], the power method with deflation [17], the method of estimating K with a subset of the data [17], the Sparse Greedy matrix Approximation (SGA) [18], the Nystrom method [21], the sparse kernel PCA method based on the probabilistic feature-space PCA concept [19], a method of using Prototype Reduction Schemes (PRS) to optimize kernel-based nonlinear subspace methods [6], and a hybrid method of applying PRS and Classifier Fusion Strategies (CFS) in succession after dividing the whole data set into small subsets [7]. The details of these schemes have been omitted in the interest of brevity.

Various dimension selection methods have been reported in the literature [8], [13]. Two representative schemes are: 1) a selection method of considering the cumulative proportion and 2) an iterative selection method of using a learning modification. Both of these methods and their advantages and disadvantages are briefly reviewed presently.

Although both of these methods are effective, they pose difficulties when applied to real-life problems. Rather, the problem of selecting optimal subspace dimensions for KNS classifiers can be seen as equivalent to searching for the best solution from an underlying solution space. Thus, we consider this problem as a *search* problem and propose utilizing AI search methods (such as the *Breadth-First Search* (BFS) and the *Depth-First Search* (DFS) methods (see [5] and [15])) to select the most appropriate dimensions. It is well-known that the choice of a heuristic function is problem dependent and, in this case, we use one which is called the *Overlapping* criterion, defined by the angle between the subspaces specified by the eigenvector columns. In other words, if the optimal subspace dimensions are used, there is no overlap between the subspaces. Thus, we propose using combinations of

the Overlapping criterion, the BFS, the DFS, and enhancements of the latter schemes to lead to dimension-selection strategies. Viewed from a computational perspective, since the optimality of our decision process is finally based on classification errors, we can say that the “Overlap Criterion” is *not an optimality* criterion, but, rather, a pruning criterion. All of these schemes have been tested for artificial and real-life benchmark data sets.

1.1 Contributions of the Paper

The main contribution of this paper is that we have reduced the problem of determining the best dimensions for KNS methods to be one of searching in an appropriately modeled solution search-space. To our knowledge, such a modeling is unique to the field of KNS classifiers. It also permits us to employ state-space search algorithms to lead to the solution desired. The second contribution of this paper is that we have proposed a good heuristic function, namely, the *Overlapping* criterion, which assists the search process. Finally, the third contribution of the paper is that we have used traditional search schemes, such as the BFS and DFS, in conjunction with the heuristic function represented by the *Overlapping matrix* to yield enhanced searching methods, which we have called the OBF and ODF methods, respectively. All these methods have also been tested² on artificial and real-life data, thus demonstrating that our hypotheses are justified. Thus, we believe that the problem of determining the best dimensions for a KNS method can be tackled in a way that has not been reported in the literature until today. However, we hasten to add that, although the paper highlights some issues associated with subspace selection, it does not address all of the numerous related issues. We view our solution as a small piece in the overall puzzle.

2 KERNEL BASED NONLINEAR SUBSPACE (KNS) METHOD

In this section, we provide a brief overview to the kernel-based nonlinear subspace method. Since the eigenvectors are computed linearly by the Principle Component Analysis (PCA), these classifiers are very effective for linear problems, but not good for nonlinear feature distributions. This has naturally motivated various developments of nonlinear PCAs, including the kernel PCA (or kPCA). The details of the kPCA are omitted (as they are essentially irrelevant), but can be found in well-known literature and also in [16] and [17].

The Kernel-based Nonlinear Subspace (KNS) method is a subspace classifier where the kPCA is employed as the specific nonlinear feature extractor. Given a set of n N -dimensional data samples, $X = (x_1, \dots, x_n)^T \in \omega_i$, and a kernel function $k(\cdot, \cdot)$, the column matrix of the orthogonal eigenvectors, $U = (U_1, \dots, U_{p_i})$, and the diagonal matrix of the eigenvalues, Λ , can be computed from the kernel matrix $K_{ij} = k(x_i, x_j)$, $1 \leq i, j \leq n$. Then, it can be shown [6] that the length of the projection of a new test pattern z onto the principal directions of U in F is

$$\|P_i\|^2 = \sum_{j=1}^{p_i} \sum_{i=1}^n (\alpha_{ji} k(x_i, z))^2 = \left\| \frac{1}{\sqrt{\Lambda}} U^T k(X, z) \right\|^2. \quad (1)$$

2. In the present paper, all of the experiments have only been performed on 2-class problems. In essence, this means that we treat the r -class problem as a hierarchy of 2-class problems. The problem of directly dealing with the r -class problem is open and is far more complex because the number of dimensions and the actual dimensions useful for comparing class i against class j may be substantially distinct from the corresponding quantities if class i is being compared against class k . We are unaware of how the general r -class problem can be solved, other than by resorting to the brute-force method that is currently recommended in the literature. Clearly, this is one “glaring” open problem.

Thus, the analogous decision rule, as that of linear subspace classifiers, classifies each z into the class on whose “class subspace” it has the longest projection as follows:

$$\forall j \neq i, \|P_i\|^2 > \|P_j\|^2 \Rightarrow z \in \omega_i. \quad (2)$$

In KNS, the subspace dimensions, $p_i, i = 1, \dots, r$, have a strong influence on the performance of subspace classifiers. In order to get a high classification accuracy, we have to select a large dimension. However, designing with dimensions that are too large leads to low performance due to overlapping of the resultant subspaces. Also, since the speed of computation for the discriminant function of subspace classifiers decreases with the dimension, the latter should be kept small.

Various dimension selection methods have been reported in the literature [8], [13]. The simplest approach is to select a global dimension to be used for all the classes. A more popular method is to select different dimensions for the various classes based on the cumulative proportion, α , which is defined as $\alpha(p_i) = \sum_{i=1}^{p_i} \lambda_i / \sum_{i=1}^n \lambda_i$.

In this method, the p_i of each class for the data sets is determined by considering the cumulative proportion, $\alpha(p_i)$. For each class, the kernel matrix, K , is computed using the available training samples for that class. The eigenvectors and eigenvalues are computed and the cumulative sum of the eigenvalues, normalized by the trace of K , is compared to a fixed number.

3 SCHEMA FOR THE PROPOSED SOLUTIONS

A heuristic-based solution for the problem of obtaining the optimal³ dimensions is necessary in light of the following conjecture.⁴

Conjecture 1. *The problem, KNS-Dim, of determining the optimal KNS dimensions is NP-hard whenever d , the dimension of the kernel space, is greater than 2.*

Informal Reasoning: The proof of this conjecture is open. Although we believe that this can be achieved by a reduction relating the KNS-Dim problem and the graph partitioning problem, the details are yet to be worked out. In the absence of more specific information about the actual kernel function and the actual training data set, there are other reasons to believe in the NP-hardness of the KNS-Dim problem. First of all, since the kernel function can be fairly arbitrary (in definition and in the real-life application domain), it is clear that it is impossible to predict a priori the values of the projections on the various dimensions. Second, the actual kernel matrix depends on the data set and not just on the function under consideration. Thus, even if we used an iterative scheme to migrate (in the solution space) from one set of dimensions to a, hopefully, “better” solution, it does not appear that we can determine a preferential search direction from any point in the solution space. Thus, it appears as if there is no formal algorithmic strategy to learn the optimal solution to the KNS-Dim problem. Thus, we proceed with the major thrust of this paper, namely, to develop techniques for obtaining approximate solutions to the problem and, to achieve this, we seek a proper heuristic function.

3. The criterion for optimality here is the “zero-one” loss on the classification error.

4. An argument from classification theory (as opposed to one from computational theory) would have probably been more appropriate here. But, as pointed out earlier, since we are not utilizing the *overlapping* criterion as an optimality criterion (but rather as a *pruning* criterion), we believe that such a result could provide the best rationale for our strategy. We are thankful to the referee who brought this to our attention.

3.1 Developing Approximate Solutions to the KNS-Dim Problem

From a computational and practical perspective, Conjecture 1 justifies the research for developing heuristic-based solutions because it appears that the optimal solution can only be obtained by an exhaustive search of the entire solution space. The computation of the exact solution by a “brute force” strategy would require searching over all possible choices of dimensions, which is clearly infeasible. Finally, as mentioned above, there seems to be no systematic way by which any partial solution can be discarded, except by some type of branch-and-bound philosophy in which a particular set of dimensions is discarded (after it is initially investigated) when its current *partial* solution is already less accurate than the *total* accuracy of another choice of dimensions.

3.1.1 Rationale for the Heuristic—The Overlapping Criterion

Motivated by the recent results⁵ of Oommen and Rueda [14], we now propose a “good” heuristic solution to the KNS-Dim problem. This solution uses a heuristic criterion that involves critical information about the specific dimensions chosen, called the *Overlap*, between the corresponding subspaces.

The solution we propose is based on a heuristic-based method that operates in two stages. For a given set of dimensions, the first stage seeks to determine whether the information contained in the subspaces characterized by the specific dimensions is superfluous. If that is the case (i.e., if the overlap between them is large), the information contained in the subspaces has a high probability of being superfluous and, so, using the mathematical principles derived in [14], it is *not* advantageous to retain these dimensions in the final solution. The heuristic looks for cases when the intersubspace overlap is minimized and this is achieved by using the *Overlapping* criterion described presently. Subsequent to this, the second phase uses this heuristic function to search the solution space by appropriately using either a BFS or a DFS.

Let $U = (U_1, \dots, U_{p_u})$ and $V = (V_1, \dots, V_{p_v})$ be two subspaces defined by the eigenvectors corresponding to the eigenvalues $\lambda_1 \geq \dots \geq \lambda_{p_u}$ and $\lambda'_1 \geq \dots \geq \lambda'_{p_v}$, respectively. First of all, observe that if U and V are column vectors of unit length, the *angle* between them is defined as $\arccos(U \cdot V)$. However, in the more general case, if U and V are matrices, the angle between them is related to the projection of one subspace onto the second. Observe that it is not necessary that the two subspaces U and V be of the same size in order to find this projection,⁶ $Projection(U, V)$. Geometrically, this is computed in terms of the angle between two hyperplanes embedded in a higher dimensional space and, in the case of subspaces, this projection between the two subspaces specified by the two matrices of column eigenvectors U and V , respectively, can be computed directly using the *subspace* function of a package like MATLAB,⁷

$$\theta = \text{subspace}(U, V), \quad (3)$$

and is a direct function of the projection of the first subspace on the second.

The angle θ of (3) gives us a measure of the amount of overlap between the two subspaces. To utilize this criterion, we *estimate* the optimal subspace dimensions as:

$$\theta(p_i, p_j) = \text{subspace}(U(p_i), U'(p_j)), \quad (4)$$

5. Unfortunately, due to space limitations, we cannot explain this result here in any detail. Informally speaking, the main thrust of the theoretical result in [14] is that, whenever we seek to find a heuristic solution for a problem, it is always advantageous to utilize heuristic functions that use “clues” which, in turn, lead to good solutions with *high probability*.

6. Although U and V do not need to be of the same size, they must, however, have the same number of rows.

7. <http://www.mathworks.com>.

where $U(p_i) = (U_1, \dots, U_{p_i})$ and $U'(p_j) = (U'_1, \dots, U'_{p_j})$ are the column eigenvectors of class i and class j , computed as in [6], and p_i and p_j are the numbers of columns, respectively. To achieve this using the *Overlapping Criterion*, we define an *Overlapping Matrix* O as:

$$O = \begin{pmatrix} \theta(1,1) & \theta(1,2) & \dots & \theta(1,p_j) \\ \theta(2,1) & \theta(2,2) & \dots & \theta(2,p_j) \\ \dots & \dots & \dots & \dots \\ \theta(p_i,1) & \theta(p_i,2) & \dots & \theta(p_i,p_j) \end{pmatrix}, \quad (5)$$

where $0 \leq O(k,l) \leq \frac{\pi}{2}$, $1 \leq k \leq p_i$, and $1 \leq l \leq p_j$.

To illustrate this, consider two subspaces described by U and U' given below:

$$U = (U_1, U_2) = \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix}, U' = (U'_1, U'_2) = \begin{pmatrix} 0 & 1 \\ 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

The *subspace* $(U(p_i), U'(p_j))$, $p_i = p_j = 2$, gives the overlapping matrix O as:

$$O = \begin{pmatrix} \frac{\pi}{2} & \frac{\pi}{4} \\ 0 & \frac{\pi}{4} \end{pmatrix}.$$

In the above, the fact that $O(1,1) = \frac{\pi}{2}$ means that two subspaces, described by two basis vectors (U_1) and (U'_1) , respectively, are orthogonal to each other. On the other hand, $O(2,1) = 0$ shows that two subspaces of (U_1, U_2) and (U'_1) are completely coincident with each other. Similarly, $O(1,2) = O(2,2) = \frac{\pi}{4}$ represents the condition that the two subspaces, constructed by (U_1) , (U'_1, U'_2) and (U_1, U_2) , (U'_1, U'_2) , respectively, are partly coincident, with an angle of $\frac{\pi}{4}$.

3.1.2 On The Overlapping Criterion

To visually demonstrate the contribution of the so-called *Overlapping Criterion*, we demonstrate the characteristics of the overlapping matrix which will be used to implement the proposed new criterion. Fig. 1a shows a 3D plot of the overlapping matrix for one of the data sets we have examined, the “Non_linear 1” set [9], where $0 \leq O(k,l) \leq \frac{\pi}{2}$, $1 \leq k \leq 10$, $1 \leq l \leq 10$. Here, the longitudinal axis is for the values of $O(k,l)$. Fig. 1b shows a contour plot of the overlapping matrix shown in Fig. 1a, where each contour line represents $\theta(k,l)$ having the same value.

In Fig. 1b, it can be observed that the total search area of $n \times n$ was reduced to a search space of dimensions $p_m \times p_m$ (where, in this case, $p_m = 10$) by employing a parameter ρ . In this case, $\rho = 0.999$. The minimum overlapping (the ridge of the mountain) occurs at the diagonal elements of $k \approx l$. This means that the possibility of selecting optimal subspace dimensions is maximum within the most inner contour line. Therefore, it can also be observed that the search area of $p_m \times p_m$ (10×10 in the figure) can be further reduced into the most inner contour area by employing the second parameter ρ' , which, in this case, is $\rho' = 0.9$. Based on these characteristics, we select the dimensions efficiently by avoiding excessive/futile searching.

4 SYSTEMATIC METHODS FOR SUBSPACE DIMENSION SELECTION

4.1 A Method Based on Overlapping and Breadth-First (OBF)

The search in any state-space is typically done using two fundamental strategies, namely, the *Breadth-First Search* (BFS) and the *Depth-First Search* (DFS), well-known in the literature [5], [15].⁸ We describe these two options below.

8. BFS and DFS methods can be called *blind-search procedures* since the order in which nodes are expanded is unaffected by the location of the goal.

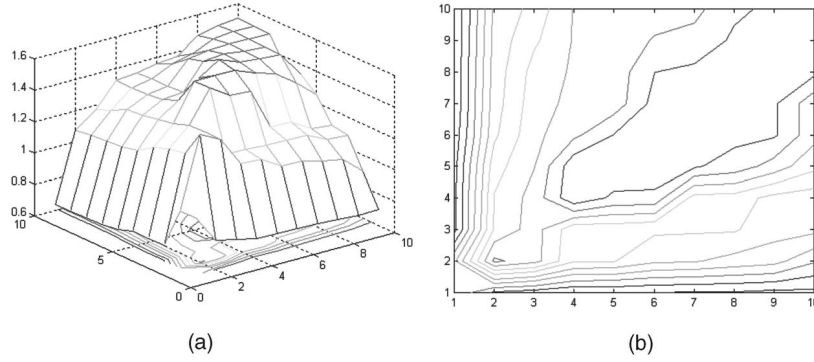


Fig. 1. A 3D plot of the overlapping matrix for the “Non_linear 1” data set, where $0 \leq O(k, l) \leq \frac{\pi}{2}$, $1 \leq k \leq 10$, and $1 \leq l \leq 10$. The details of the figure are found in the text. A contour plot of the overlapping matrix shown in Fig. 1a is given in Fig. 1b.

The problem of selecting a subspace dimension (p_i, p_j) for a given n -dimensional application is one of selecting pair (k, l) from the $n \times n$ integer space. A “hybrid” procedure based on utilizing the Overlapping criterion, in conjunction with a BFS (referred to here as *OBF*), is formalized below, where the training set is given by T and the p_i -dimensional subspace is given by $U(p_i)$.

1. Compute the kernel matrix, K , with T for each class, i . From K , compute the eigenvectors, $U = (U_1, \dots, U_{p_i})$, and the eigenvalues, $\lambda_1 \geq \dots \geq \lambda_{p_i}$;
2. Compute the overlapping matrix $O(k, l)$, $1 \leq k \leq p_i$, $1 \leq l \leq p_j$, for a class i and a class j with the constraint that $O(k, l) \leq \rho$. Then, set $p_m = \text{argmax}\{\max_l\{\max_k\{O(k, l)\}\}\}$;
3. Evaluate the accuracy of the subspace classifier defined by $U(k)$ and $U'(l)$ by traversing all values of l and k , $1 \leq k \leq p_m$, $1 \leq l \leq p_m$;
4. Report the dimension of the subspace by selecting (p_i, p_j) with the highest accuracy.

In Step 2, p_m is determined by the parameter, ρ , which is a parameter used to prune the upper area of the search space so as to reduce the corresponding CPU-time. However, since Step 2 is computed only once, it turns out that almost all of the processing CPU-time is utilized in Step 3, the evaluation step. So, to further reduce the processing, Step 3 can be replaced by the following:

3. If $O(k, l) > \rho'$, evaluate the accuracy of the classifiers defined by $U(k)$ and $U'(l)$ with T .

In other words, by limiting the possible values of ρ , we can prune off the *upper* search area, which corresponds to excessively large values of k and l , and, by using the constraint on ρ' , we can also prune off the *lower* search area, which corresponds to values of k and l which are too small. We refer to this *Constrained OBF* as *COBF* in the following sections.

4.2 A Method Based on Overlapping and Depth-First (ODF)

Since the OBF is time-consuming, we propose another approach in which the optimal or near-optimal subspace dimensions are selected by searching the solution space in a specified direction, namely, that which is determined by using the overlapping matrix. This scheme can reduce the searching time significantly. The procedure can be formalized as follows, where the training set is given by T and the p_i dimensional subspace is represented in $U(p_i)$:

1. Compute the kernel matrix K with T for each class, i . From the K , compute the eigenvectors, $U = (U_1, \dots, U_{p_i})$, and the eigenvalues, $\lambda_1 \geq \dots \geq \lambda_{p_i}$;

9. p_m , which becomes the upper bound for the indices, is set merely as a computational convenience.

2. Compute the overlapping matrix $O(k, l)$, $1 \leq k \leq p_i$, $1 \leq l \leq p_j$, for class i and class j with the constraint that $O(k, l) \leq \rho$. Then, set $p_m = \text{argmax}\{\max_l\{\max_k\{O(k, l)\}\}\}$;
3. Do the following by increasing q from *unity* to p_m in steps of *unity* per epoch:
 - a. Set $(k, l) = \text{argmax}\{O(q-1, q), O(q, q), O(q, q-1)\}$ or $(k, l) = (1, 1)$ if $q = 1$;
 - b. Evaluate the accuracy of the subspace classifiers defined by $U(k)$ and $U'(l)$ with T .
4. Report the dimensions of the subspaces by selecting those that yield the highest accuracy.

As in the case of the OBF, almost all of the processing CPU-time of ODF is utilized in executing Step 3, the evaluation step. So, again, to further reduce the processing time, the upper and lower search areas can be pruned. Rather than reiterate the same steps, we merely state that such a refined scheme will be referred to as the *Constrained ODF* or *CODF* in the following sections.

We conclude this section with a final remark about the algorithms OBF and ODF, which both take pairs of subspaces as their inputs. Ideally, each subspace should be optimized against all the other classes. But, since this would be computationally exorbitant, in our schemes, the i and j in Step 2 are selected randomly and among them we determine the optimal or near-optimal dimensions based on the resulting classification errors.

5 EXPERIMENTAL RESULTS

The proposed methods *COBF* and *CODF* have been rigorously tested and compared with many conventional KNSs on both “artificial” and “real-life” data sets. In our experiments, the six artificial data sets, “Non_normal 1,2,3” and “Non_linear 1,2,3,” were generated with different sizes of testing and training sets of cardinality 50, 500 and 5,000, respectively. The data sets “Iris2,”¹⁰ “Arrhythmia” (in short, “Arrhy”), and “Adult4,” which are real benchmark data sets, are cited from the UCI Machine Learning Repository [3]. In each case, the vectors were normalized within the range $[-1, 1]$ using their standard deviations. Also, for every class j , the data set for the class was randomly split into two subsets of equal size for training and testing and their roles were later interchanged.

The data set named “Non_normal” was generated from a mixture of four eight-dimensional Gaussian distributions as follows: $p_1(x) = \frac{1}{2}N(\mu_{11}, I_8) + \frac{1}{2}N(\mu_{12}, I_8)$ and $p_2(x) = \frac{1}{2}N(\mu_{21}, I_8) + \frac{1}{2}N(\mu_{22}, I_8)$, where $\mu_{11} = [0, 0, \dots, 0]$, $\mu_{12} = [6.58, 0, \dots, 0]$, $\mu_{21} = [3.29, 0, \dots, 0]$,

10. The “Iris” data set consisted of three classes: Setosa, Versicolor, and Virginica. However, since the subset of Setosa samples is completely separated from the others, it is not difficult to classify it from the other two. Therefore, we have opted to employ a modified set.

TABLE 1
The Experimental Results of the Proposed Method for the Artificial and Real-Life Data Sets

Datasets	Methods	Acc	(p_1, p_2)	t_1	t_2	Parameters
Non.n1	CCM	95.00	(2, 2) (2, 2)	0.18	0.28	$\Delta\alpha \leq 0.001$
	CPE	95.00	(1, 1) (1, 1)	5.65	0.25	$k = 0.98; k_0 = 0.98; \Delta k = 0.001$
	OBf	95.00	(1, 1) (1, 1)	23.15	0.25	$\rho = 0.99; p_m = 6, 11$
	ODf	95.00	(1, 1) (1, 1)	2.84	0.26	$\rho = 0.99; p_m = 6, 11$
	COBF	50.00	(5, 5) (6, 9)	5.76	0.32	$\rho = 0.99; \rho' = 0.7; p_m = 6, 11$
	CODf	48.00	(5, 5) (6, 6)	1.82	0.31	$\rho = 0.99; \rho' = 0.7; p_m = 6, 11$
Non.n2	CCM	94.80	(2, 2) (2, 2)	16.59	23.63	$\Delta\alpha \leq 0.001$
	CPE	94.80	(2, 2) (1, 1)	510.18	23.64	$k = 0.980; k_0 = 0.98; \Delta k = 0.001$
	OBf	94.80	(2, 2) (1, 1)	1,745.96	23.36	$\rho = 0.99; p_m = 3, 11$
	ODf	94.80	(2, 2) (1, 1)	201.33	23.49	$\rho = 0.99; p_m = 3, 11$
	COBF	94.80	(2, 2) (2, 2)	1,221.93	25.24	$\rho = 0.99; \rho' = 0.9; p_m = 3, 11$
	CODf	94.80	(2, 2) (2, 2)	185.25	24.45	$\rho = 0.99; \rho' = 0.9; p_m = 3, 11$
Non.n3	CCM	94.86	(2, 2) (2, 2)	5,550.00	2,384.30	$\Delta\alpha \leq 0.001$
	CPE	94.86	(2, 2) (2, 2)	54,244.00	2,340.50	$k = 0.991; k_0 = 0.98; \Delta k = 0.001$
	OBf	94.86	(2, 2) (2, 2)	44,009.50	2,324.50	$\rho = 0.99; p_m = 2, 5$
	ODf	94.86	(2, 2) (2, 2)	17,127.00	2,358.50	$\rho = 0.99; p_m = 2, 5$
	COBF	94.86	(2, 2) (2, 2)	28,629.40	2,380.00	$\rho = 0.99; \rho' = 0.7; p_m = 2, 5$
	CODf	94.86	(2, 2) (2, 2)	14,813.00	2,377.50	$\rho = 0.99; \rho' = 0.7; p_m = 2, 5$
Non.I1	CCM	78.00	(5, 4) (4, 4)	0.16	0.28	$\Delta\alpha \leq 0.00001$
	CPE	90.00	(3, 3) (3, 3)	3.34	0.26	$k = 0.9995; k_0 = 0.999; \Delta k = 0.0001$
	OBf	85.00	(5, 5) (3, 3)	156.05	0.27	$\rho = 0.999; p_m = 9, 25$
	ODf	85.00	(5, 5) (3, 3)	7.69	0.26	$\rho = 0.999; p_m = 9, 25$
	COBF	85.00	(5, 5) (3, 3)	130.86	0.27	$\rho = 0.999; \rho' = 0.9; p_m = 9, 25$
	CODf	85.00	(5, 5) (3, 3)	7.54	0.27	$\rho = 0.999; \rho' = 0.9; p_m = 9, 25$
Non.I2	CCM	84.70	(5, 5) (5, 4)	16.27	24.06	$\Delta\alpha \leq 0.00001$
	CPE	84.70	(3, 3) (3, 3)	295.33	23.85	$k = 0.9994; k_0 = 0.999; \Delta k = 0.0001$
	OBf	89.70	(5, 5) (5, 5)	5,053.90	24.55	$\rho = 0.999; p_m = 17, 8$
	ODf	91.60	(5, 5) (1, 1)	416.11	23.88	$\rho = 0.999; p_m = 17, 11$
	COBF	89.70	(5, 5) (5, 5)	3,763.40	24.25	$\rho = 0.999; \rho' = 0.9; p_m = 17, 8$
	CODf	89.70	(5, 5) (5, 5)	344.23	24.11	$\rho = 0.999; \rho' = 0.9; p_m = 17, 8$
Non.I3	CCM	89.70	(5, 5) (5, 5)	5,523.40	2,403.80	$\Delta\alpha \leq 0.001$
	CPE	85.73	(3, 3) (3, 3)	33,232.00	2,361.00	$k = 0.9910; k_0 = 0.999; \Delta k = 0.0001$
	OBf	86.95	(5, 5) (3, 3)	430,819.50	2,413.50	$\rho = 0.999; p_m = 17, 3$
	ODf	86.95	(5, 5) (3, 3)	36,999.00	2,362.00	$\rho = 0.999; p_m = 17, 3$
	COBF	86.95	(5, 5) (3, 3)	1,294,357.00	2,373.50	$\rho = 0.999; \rho' = 0.9; p_m = 33, 3$
	CODf	86.95	(5, 5) (3, 3)	33,366.00	2,375.00	$\rho = 0.999; \rho' = 0.9; p_m = 17, 3$
Iris2	CCM	89.00	(1, 1) (1, 1)	0.17	0.27	$\Delta\alpha \leq 0.01$
	CPE	90.00	(3, 3) (2, 2)	5.51	0.27	$k = 0.9950; k_0 = 0.98; \Delta k = 0.001$
	OBf	94.00	(5, 6) (5, 6)	223.99	0.27	$\rho = 0.998; p_m = 24, 25$
	ODf	94.00	(6, 6) (11, 11)	11.54	0.29	$\rho = 0.998; p_m = 24, 25$
	COBF	94.00	(6, 6) (10, 11)	129.99	0.31	$\rho = 0.998; \rho' = 0.9; p_m = 24, 25$
	CODf	94.00	(6, 6) (11, 11)	11.09	0.31	$\rho = 0.998; \rho' = 0.9; p_m = 24, 25$
Arrhy	CCM	99.56	(73, 103) (75, 102)	5.22	40.08	$\Delta\alpha \leq 0.0001$
	CPE	98.90	(67, 66) (74, 71)	827.57	23.63	$k = 0.980; k_0 = 0.98; \Delta k = 0.001$
	OBf	98.46	(1, 1) (7, 7)	4,997.55	6.75	$\rho = 0.998; p_m = 21, 30$
	ODf	98.46	(1, 1) (7, 7)	197.92	6.62	$\rho = 0.998; p_m = 21, 30$
	COBF	97.13	(6, 9) (7, 7)	2,455.35	6.70	$\rho = 0.998; \rho' = 0.9; p_m = 21, 30$
	CODf	97.13	(9, 9) (7, 7)	198.70	6.77	$\rho = 0.998; \rho' = 0.9; p_m = 21, 30$
Adult4	CCM	87.53	(9, 9) (9, 9)	10,688.50	1,852.00	$\Delta\alpha \leq 0.001$
	CPE	87.53	(9, 9) (9, 9)	46,773.50	1,799.50	$k = 0.9960; k_0 = 0.98; \Delta k = 0.001$
	OBf	95.07	(12, 6) (5, 1)	477,605.00	1,760.00	$\rho = 0.998; p_m = 12, 20$
	ODf	87.53	(9, 9) (9, 9)	39,088.50	1,738.50	$\rho = 0.998; p_m = 12, 20$
	COBF	95.08	(12, 6) (10, 5)	305,165.00	1,645.00	$\rho = 0.998; \rho' = 0.9; p_m = 12, 20$
	CODf	87.53	(9, 9) (9, 9)	38,917.00	1,826.50	$\rho = 0.998; \rho' = 0.9; p_m = 12, 20$

(See the text for the meanings of the columns.) The final column gives the experimental Parameters, where $\Delta\alpha = \alpha(p_i + 1) - \alpha(p_i)$, k , k_0 , Δk , ρ , ρ' , and p_m are those of the corresponding algorithm, explained in the text, respectively. In the columns of (p_1, p_2) and p_m , the first one is for the training sets, and the second one is for the test sets.

and $\mu_{22} = [9.87, 0, \dots, 0]$. In these expressions, I_8 is the eight-dimensional Identity matrix. The data set named “Non_linear” (in short, “Non_l”) [9], which has a strong nonlinearity at its boundary, was generated artificially from a mixture of four variables, as shown in [9].

The “Arrhythmia” data set contains 279 attributes, 206 of which are real-valued and the rest are nominal. In our experiments, the nominal features were replaced by zeros. The aim of the pattern recognition exercise was to distinguish between the presence or absence of cardiac arrhythmia and to classify the feature into one of the 16 groups. In our case, in the interest of uniformity, we merely attempted to classify the total instances into two categories, namely, “normal” and “abnormal.”

The “Adult4” data set was extracted from a census bureau database [3]. The aim of the pattern recognition task here is to

separate people by incomes into two groups; in the first group, the salary is more than 50K dollars and, in the second group, the salary is less than or equal to 50K dollars. Each sample vector has 14 attributes. Some of the attributes, such as the age, hours-per-week, etc., are continuous numerical values. The others, such as education, race, etc., are nominal symbols. In this case, the total number of sample vectors is 33,330. Due to time considerations, we randomly selected 8,336 samples—approximately 25 percent of the set.

The kernel function employed in the experiments was the polynomial kernel, with the intercept and the degree of the polynomial given as $k(i, j) = (1 + x'_i x_j)^2$.

We report below the runtime characteristics of the proposed algorithms for the artificial and real-life data sets. Table 1 shows the experimental results for all of the artificial and real-life data sets.

Here, the abbreviations CCM, CPE, OBF, and ODF are the Conventional Cumulative Method, the simple method based on Cumulative Proportion and Evaluation,¹¹ the method based on Overlapping criterion and Breadth-First search, and the method based on Overlapping criterion and Depth-First search, respectively. Also, the COBF and CODF are the constrained versions of the OBF and ODF, respectively. In the table, each result is the averaged value of the training and the test sets, respectively. Also, to be objective in this task, we compared the proposed methods and the traditional schemes¹² using three criteria (averaged for the two data sets), namely, the classification accuracy (percent), Acc ; the selected subspace dimensions, (p_1, p_2) ; and t_1 and t_2 , which are the processing CPU-times (in seconds) for their dimension selections and classifications, respectively. In each case, although ρ and ρ' were set by trial-and-error, their actual values did not make much of a difference as long as they were relatively close to unity.

With regard to real-life data, consider the results of the data set captioned *Arrhy*. The classification accuracies, Acc , of the CCM, CPE, OBF, ODF, COBF, and CODF methods, are 99.56, 98.90, 98.46, 98.46, 97.13, and 97.13 (percent), respectively. The required selection CPU-times, t_1 , are 5.22, 827.57, 4,997.55, 197.92, 2,455.35, and 198.70 seconds, and the obtained two subspace dimension sets are (73, 103) (75, 102), (67, 66) (74, 71), (1, 1) (7, 7), (1, 1) (7, 7), (6, 9) (7, 7), and (9, 9) (7, 7) with the methods. The classification times, t_2 , are 40.08, 23.63, 6.75, 6.62, 6.70, and 6.77 seconds, respectively. Again, it is clear that an optimal or near-optimal subspace dimension can be selected systematically and with extreme efficiency by using the CODF method. These comments can also be made about the results obtained from the other data sets too.

In conclusion, it appears that the CODF is uniformly superior to the others in terms of the classification accuracy, Acc , while requiring a marginal additional "dimension selection" time, t_1 .

6 CONCLUSIONS AND FUTURE WORK

In this paper, we propose a new strategy for systematically and efficiently selecting optimal or near-optimal subspace dimensions for a KNS classifier using well-known, AI-based *search* strategies. In order to achieve this, we have also proposed a new heuristic function, called the *Overlapping Criterion*, and given a rationale for using it based on a recent result by Oommen and Rueda [14]. We have suggested that, by using this heuristic function, the search time required to determine the best solution can be reduced significantly.

The proposed methods have been tested on artificial and real-life benchmark data sets and compared with conventional schemes. The experimental results for both data sets demonstrate that one of the newly proposed schemes, a Constrained version which utilizes the Overlapping matrix and Depth-First strategy (CODF), can select subspace dimensions systematically and very efficiently.

The problems of determining the best type of kernel function and of optimizing the corresponding parameters of the function are still open. Also, the possible relevance of the overlapping criterion and the use of heuristic search methods to other Kernel-based algorithms (and to not just subspace classifiers) remains open. Most

importantly, the entire problem of analyzing the efficiency of the heuristic function (the "Overlapping criterion") remains open. However, we believe that such an analysis will be *far from trivial* because it will be both problem dependent and also dependent on the *distribution* of the points encountered in the *mapped space* and not the distribution of the points in the original feature space.

ACKNOWLEDGMENTS

The work of Sang-Woon Kim was done while visiting at Carleton University, Ottawa, Canada. He was partially supported by KOSEF, the Korea Science and Engineering Foundation. B. John Oommen was partially supported by NSERC, the Natural Sciences and Engineering Research Council of Canada. A preliminary version of some of the results of this paper was presented at AI '04, the 2004 Australian Joint Conference on Artificial Intelligence, in Perth, Australia, in December 2004.

REFERENCES

- [1] D. Achlioptas and F. McSherry, "Fast Computation of Low-Rank Matrix Approximations," *Proc. 33rd Ann. ACM Symp. Theory of Computing (STOC '01)*, pp. 611-618, 2001.
- [2] D. Achlioptas, F. McSherry, and B. Schölkopf, "Sampling Techniques for Kernel Methods," *Advances in Neural Information Processing Systems 14*, pp. 335-342, 2002.
- [3] C.L. Blake and C.J. Merz, "UCI Repository of Machine Learning Databases," Dept. of Information and Computer Science, Univ. of California, Irvine, <http://www.ics.uci.edu/mllearn/MLRepository.html>, 1998.
- [4] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, second ed. San Diego, Calif.: Academic Press, 1990.
- [5] L.N. Kanal, "Problem-Solving Models and Search Strategies for Pattern Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 1, no. 2, pp. 193-201, 1979.
- [6] S.-W. Kim and B.J. Oommen, "On Using Prototype Reduction Schemes to Optimize Kernel-Based Nonlinear Subspace Methods," *Pattern Recognition*, vol. 37, no. 2, pp. 227-239, 2004.
- [7] S.-W. Kim and B.J. Oommen, "On Using Prototype Reduction Schemes and Classifier Fusion Strategies to Optimize Kernel-Based Nonlinear Subspace Methods," submitted for publication.
- [8] J. Laaksonen and E. Oja, "Subspace Dimension Selection and Averaged Learning Subspace Method in Handwritten Digit Classification," *Proc. Int'l Conf. Artificial Neural Networks (ICANN '96)*, pp. 227-232, 1996.
- [9] E. Maeda and H. Murase, "Multi-Category Classification by Kernel Based Nonlinear Subspace Method," *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing (ICASSP '99)*, 1999.
- [10] K. Maeda and S. Watanabe, "A Pattern Matching Method with Local Structure (in Japanese)," *IEICE Trans. Information & Systems*, vol. J68-D, no. 3, pp. 345-352, Mar. 1985.
- [11] K.R. Müller, S. Mika, G. Ratsch, K. Tsuda, and B. Schölkopf, "An Introduction to Kernel-Based Learning Algorithms," *IEEE Trans. Neural Networks*, vol. 12, no. 2, pp. 181-201, Mar. 2001.
- [12] N.J. Nilsson, *Problem-Solving Methods in Artificial Intelligence*. McGraw-Hill, 1971.
- [13] E. Oja, *Subspace Methods of Pattern Recognition*. Research Studies Press, 1983.
- [14] B.J. Oommen and L. Rueda, "A Formal Analysis of Why Heuristic Functions Work," *The Artificial Intelligence J.*, to appear.
- [15] E. Rich and K. Knight, *Artificial Intelligence*, second ed. McGraw-Hill, 1991.
- [16] B. Schölkopf, S. Mika, C.J.C. Burges, P. Knirsch, K.R. Müller, G. Ratsch, K. Tsuda, and A.J. Smola, "Input Space Versus Feature Space in Kernel-Based Methods," *IEEE Trans. Neural Networks*, vol. 10, no. 5, pp. 1000-1016, Sept. 1999.
- [17] B. Schölkopf, A.J. Smola, and K.R. Müller, "Nonlinear Component Analysis as a Kernel Eigenvalue Problem," *Neural Computation*, vol. 10, no. 5, pp. 1299-1319, 1998.
- [18] A.J. Smola and B. Schölkopf, "Sparse Greedy Matrix Approximation for Machine Learning," *Proc. Int'l Conf. Machine Learning (ICML '00)*, pp. 911-918, 2000.
- [19] M. Tipping, "Sparse Kernel Principal Component Analysis," *Advances in Neural Information Processing Systems 13*, Cambridge, Mass.: MIT Press, 2001.
- [20] K. Tsuda, "Subspace Method in the Hilbert Space (in Japanese)," *IEICE Trans. Information & Systems*, vol. J82-D-II, no. 4, pp. 592-599, Apr. 1999.
- [21] C. Williams and M. Seeger, "Using the Nystrom Method to Speed Up Kernel Machines," *Advances in Neural Information Processing Systems*, vol. 13, 2001.

11. In the case of the CCM, we randomly selected a p_i as the subspace dimension based on the cumulative proportion and, in the case of the CPE, which is one of the systematic methods based on the conventional philosophies, we selected p_i as the subspace dimension obtained by considering classification errors for candidate dimensions as suggested in the literature [6].

12. The reader will observe that, although the accuracies of all of the methods are almost the same, the time required for the CCM (for example) is always less. The reason for this is that the results reported for the CCM are those obtained by randomly selecting the dimensions from a small set of possible dimension values close to the optimal. Thus, the reported CCM time does not include any search in the "dimension space." In CODF, on the other hand, we select the dimensions systematically using our heuristic criterion, but without any a priori knowledge of the "dimension space." Indeed, the result of the CCM is presented as a reference for the classification accuracy.