Growing Subspace Pattern Recognition Methods and Their Neural-Network Models

M. Prakash and M. Narasimha Murty

Abstract— **In statistical pattern recognition, the decision of which features to use is usually left to human judgment. If possible, automatic methods are desirable. Like multilayer perceptrons, learning subspace methods (LSM's) have the potential to integrate feature extraction and classification. In this paper, we propose two new algorithms, along with their neural-network implementations, to overcome certain limitations of the earlier LSM's. By introducing one cluster at a time and adapting it if necessary, we eliminate one limitation of deciding how many clusters to have in each class by trial-and-error. By using the principal component analysis neural networks along with this strategy, we propose neural-network models which are better in overcoming another limitation, scalability. Our results indicate that the proposed classifiers are comparable to classifiers like the multilayer perceptrons and the nearest-neighbor classifier in terms of classification accuracy. In terms of classification speed and scalability in design, they appear to be better for large-dimensional problems.**

Index Terms— **Subspace methods, learning methods, neural networks, constructive architectures, character recognition.**

I. Introduction

In this paper, we propose two new learning subspace pattern recognition (PR) methods along with their artificial neural network (ANN) implementations. In statistical pattern recognition and in neural networks, a separate phase of feature extraction is necessary because of the *curse of dimensionality* [1], [2]. Feedforward neural networks have the potential to integrate feature extraction (internal representation) and classification [3]–[5]. However, these attempts are far fewer compared to the attempts where features are extracted outside the network and are fed as inputs to the network [6], [7]. While discovering good features may be difficult right at the beginning, learning may be used to improve the goodness of the features chosen on some basis [8]. The two disciplines of PR and ANN's have benefited from each other [9]. Learning subspace methods (LSM's) also integrate feature extraction and classification.

Watanabe and Pakvasa [10] suggested the subspace pattern recognition method (SPRM) as a new approach to classifying and representing patterns given as elements of a vector space. Here, each class is represented by a subspace spanned by a group of basis vectors—the orthogonal components obtained by the principal component analysis. One drawback here is that the features (basis vectors) obtained for one class are not dependent on the features of the other classes. Improvements to the basic SPRM can be put under the following categories: 1) methods based on weighted orthogonal projections and 2) methods based on the rotation of subspaces. There have been many attempts of associating weights to the basis vectors in the computation of the orthogonal projections [11], [12]. By adapting the weights, these learning methods implicitly integrate feature extraction and classification. The LSM [13] and its variants [11], [14], [15] belong to the second category. The representations (i.e., the subspaces or the orthonormal features/PC's) get *adapted* in the process of rotation. Thus, the LSM integrates feature extraction and classification.

The learning methods have been successfully applied in practice on a variety of applications including speech recognition [13], [16], texture identification [17], and character recognition [12]. These methods are fast and exhibit reasonably robust behavior with respect to convergence during learning. At the time of classification, they involve only dot products and hence are extremely fast. From the point of view of computational power, two properties are associated with all the subspace methods: 1) the classification of a pattern x is based solely on its direction and does not depend on the magnitude of x and 2) the decision surfaces are quadratic. For some problems, highly nonlinear decision surfaces are required. Recently, a piecewise approximation has been adopted in the extended ALSM [18] to overcome this problem. However, this method is not scalable to large-dimensional problems. Also, the number of clusters in each of the classes and the number of basis vectors in each cluster have to be found out by trial-and-error. Constructive neural networks like the cascade correlation [19] and the growing cell structures (GCS) [20] could eliminate similar problems from multilayer perceptrons and the radial basis function neural networks, respectively, by introducing one node/unit at a time based on the misclassifications. In this paper, we propose LSM's that are good at scalability and that eliminate the possible limitation of quadratic decision surfaces on the lines of such constructive neural networks.

II. Extended Subspace Methods of Pattern Recognition

The subspace methods of classification are decision-theoretic pattern recognition methods where the primary model for a class is a linear subspace of the Euclidean pattern space [10]. Any set of m linearly independent vectors $u_1, \cdots,$ and u_m in R^n (with $m < n$) spans a subspace, L: $L = L(u_1, \cdots, u_m) = \{x/x = \sum_{i=1}^{m} \alpha_i u_i$ for some scalars $\alpha_1, \cdots, \alpha_m\}$. The basic operation on a subspace is a projection of a vector, and is given by $\hat{x} = Px$, where the projection matrix P of L is given

Manuscript received December 19, 1995; revised July 11, 1996.

The authors are with the Department of Computer Science and Automation, Indian Institute of Science, Bangalore 560 012, India.

Publisher Item Identifier S 1045-9227(97)00228-2.

by $P = A(A^tA)^{-1}A^t, A = [u_1, \cdots, u_m]$. $\tilde{x} = x - \hat{x}$ is the corresponding orthogonal residual. Given K pattern classes $\omega^1, \cdots, \omega^K$, each represented by its subspace $L^i = L(u_1^i, \cdots, u_{m^i}^i)$, the classification rule is

$$\text{if } \delta(x, L^i) > \delta(x, L^j), \forall j \neq i, \text{ then classify } x \text{ into } \omega^i \tag{1}$$

where $\delta(x, L^j)$, the squared orthogonal projection distance, is defined as

$$\delta(x, L^j) = 0\begin{cases} x^tP^jx & \text{when } u_1^j, \cdots, u_{m^j}^j \\ & \quad\text{are nonorthonormal} \\ \displaystyle\sum_{i=1}^{m^j}(x^tu_i^j)^2 & \text{when } u_1^j, \cdots, u_{m^j}^j \\ & \quad\text{are orthonormal.} \end{cases} \tag{2}$$

In the extended averaged learning subspace method (EALSM), each class is represented using one or more clusters. Let $C_1^i, \cdots, C_{n^i}^i$ represent the n^i clusters of class ω^i. Let $L^{i,j} = L(u_1^{i,j}, \cdots, u_{m^{i,j}}^{i,j})$ and $P^{i,j}$ be the subspace and the projection matrix, respectively, corresponding to the jth cluster of the ith class. The classification rule given by (1) and (2) is modified as follows:

$$\begin{aligned} &\text{if } \delta(x, L^{i,j}) > \delta(x, L^{k,l}), \quad \forall (i,j) \neq (k,l) \\ &\qquad \text{then classify } x \text{ into } (\omega^i, C_j^i) \end{aligned} \tag{3}$$

where $\delta(x, L^{i,j})$, the squared orthogonal projection distance (SOPD) on the jth cluster of the ith class, is defined as

$$\delta(x, L^{i,j}) = \begin{cases} x^tP^{i,j}x & \text{when } u_1^{i,j}, \cdots, u_{m^{i,j}}^{i,j} \\ & \quad\text{are nonorthonormal} \\ \displaystyle\sum_{k=1}^{m^{i,j}}(x^tu_k^{i,j})^2 & \text{when } u_1^{i,j}, \cdots, u_{m^{i,j}}^{i,j} \\ & \quad\text{are orthonormal.} \end{cases} \tag{4}$$

From (3) and (4), we see that the classification is independent of magnitude as in the SPRM. The decision surfaces in subspace methods are usually quadratic. In the special case of equal number of basis vectors, the decision surfaces are linear. By having many clusters in each class, and same number of basis vectors in all clusters, these extended methods obtain a piecewise linear decision boundary. In case of misclassifications, the EALSM involves the notion of a correct cluster. When (3) and (4) are used, we can find out the wrong class and the wrong cluster to which a sample is assigned. The sample has only a correct class label. The correct cluster is defined simply as that cluster with maximum SOPD among all the clusters belonging to the correct class. The subspaces pass through the origin in the EALSM. Although a better fit would be obtained if the linear manifolds pass through the cluster centers, we have not considered it as neural-network implementations are available for obtaining principal components corresponding to only correlation matrices.

III. Growing Subspace Method

The drawbacks of the EALSM are 1) it is not good at scalability and does not have a neural implementation and 2) the number of clusters for all classes has to be determined only by trial and error. Here, we propose the growing subspace method (GSM) to overcome these drawbacks. Like in the EALSM, we use several clusters to represent each class here too. Also, we use the same classification rule as given by (3) and (4).

The GSM starts with one cluster per class and keeps on growing the ensemble of clusters by introducing one cluster at a time. A new cluster can be introduced in a class only when 1) the number of clusters in this class is less than the allowable number of clusters, MAXCLUSTERS, and 2) the number of patterns of any one cluster of this class that are misclassified into another cluster of any other class exceeds a minimum number of patterns, MINPATS. This process of growing terminates when 1) all the patterns are correctly classified or 2) the number of misclassified patterns is less than MINPATS for every class where a new cluster can be introduced. Once a cluster is introduced, it remains fixed from then on.

At any stage, introducing and fixing of a cluster in the GSM proceeds as follows. Only misclassified patterns at that stage govern the introduction of a cluster. We use the notion of a correct class, correct cluster, wrong class, and wrong cluster as described in the previous section. For every quadruple (i, j, k, l), we obtain the number of misclassified patterns for which the correct class, correct cluster, wrong class, and wrong cluster are i, j, k, and l, respectively. Let the quadruple (p, q, r, s) correspond to the maximum number of misclassified patterns. Only classes for which the number of clusters is less than the maximum number of clusters allowed are considered here. If the number of misclassified samples is less than MINPATS, the entire procedure terminates. Otherwise, a new subspace is created by taking the first required number of PC's of the correlation matrix obtained using the misclassified samples corresponding to (p, q, r, s). This subspace is associated with a new cluster which is associated with the class p.

In the EALSM, one or two correlation matrices have to be updated for every pattern presentation during every iteration. Also, at the end of an iteration, eigenvectors for all the clusters have to be obtained. The scalability of the growing methods compared to the EALSM arises out of the following three factors: 1) their neural-network implementations; 2) in case of extraction of PC's using algebraic techniques—one correlation matrix will be updated *only* for those patterns for which the current cluster is either the correct cluster or the wrong cluster; eigenvectors of only this matrix will be obtained at the end of an iteration; and 3) for both neural network and algebraic implementation to get PC's—since a cluster remains fixed after its introduction, we can store the orthogonal projection distances for all the patterns for all but one cluster. These will save considerable computation time.

IV. Growing Learning Subspace Method

The LSM's adapt the subspaces such that the recognition accuracy increases. Hence, after introducing a new cluster as in the GSM, by adapting the new cluster/subspace before fixing it, we can hope to improve the recognition accuracy at that stage. Thus, the total number of clusters is likely to be less in this approach. Also, this may have a better regularization effect. Our aim here is to investigate the influence such adaptation has on the total number of clusters and the generalization capability. We call this variation as the growing learning subspace method (GLSM).

Adaptation of a subspace is done on the lines of the Hebbian learning subspace method (HLSM) [12]. By associating different weights with different basis vectors, and by adapting these weights, the classification accuracy can be improved. In the GLSM, we adapt the weights of the new cluster. The classification rule used in the GSM and the EALSM is modified as

$$\begin{aligned} &\text{if } \delta(x, L^{i,j}) > \delta(x, L^{k,l}), \quad \forall (i,j) \neq (k,l), \\ &\qquad \text{then classify } x \text{ into } (\omega^i, C^i_j) \end{aligned} \tag{5}$$

where $\delta(x, L^{i,j})$, the weighted squared orthogonal projection distance (WSOPD) on the jth cluster of the ith class, is defined as

$$\delta(x, L^{i,j}) = \begin{cases} x^t P^{i,j} x & \text{when } u^{i,j}_1, \cdots, u^{i,j}_{m^{i,j}} \\ & \quad \text{are nonorthonormal} \\ \displaystyle\sum_{k=1}^{m^{i,j}} w^{i,j}_k (x^t u^{i,j}_k)^2 & \text{when } u^{i,j}_1, \cdots, u^{i,j}_{m^{i,j}} \\ & \quad \text{are orthonormal.} \end{cases} \tag{6}$$

Let the new cluster be the jth cluster of the ith class. The weights of the new cluster are modified according to the following equations for every pattern x that is presented:

$$\delta w^{i,j}_k = \eta^+ \xi^{i,j}_k V^{i,j} \tag{7}$$

$$\delta w^{i,j}_k = \eta^- \xi^{i,j}_k V^{i,j} \tag{8}$$

$$\delta w^{i,j}_k = \eta \xi^{i,j}_k V^{i,j} \tag{9}$$

where $\xi^{i,j}_k = (x^t u^{i,j}_k)^2$ and $V^{i,j} = \Sigma_{k=1}^{m^{i,j}} w^{i,j}_k \xi^{i,j}_k$. η, η^+, and η^- are small positive constants with τt usually taken to be much smaller than the others. In case of misclassification, (8) is used when $(i, j) =$ (correct class, correct cluster) and (9) is used when $(i, j) =$ (wrong class, wrong cluster). In case of correct classification, (9) is used when $(i, j) =$ (correct class, correct cluster).

V. A Neural-Network Model for the Growing Subspace Methods

The entire neural network is depicted in Fig. 1(a), (b) and (c). Each of the subnets shown in Fig. 1(a) has the form shown in Fig. 1(b). Each of the subsubnets shown in Fig. 1(b) has the form shown in Fig. 1(c). Fig. 1(c) shows the subsubnet corresponding to the jth cluster of class i ($subsubnet^{i,j}$). The number of nodes in the first, second, and third layers are $n, m^{i,j}$, and $m^{i,j}$, respectively. The nodes in the second layer are linear. The output of a node k in this layer corresponds to $x * u^{i,j}_k$. These outputs are simply squared by the corresponding nodes in the third layer. The single node in the last layer is also linear, and thus its output corresponds to the WSOPD $V^{i,j}$.

Fig. 1(b) shows the n^i subsubnets corresponding to the n^i clusters of class i. The outputs of these subsubnets go into a modified winner-take-all neural network (MWTANN). The output of the MWTANN, z^{i,j^i}, represents V^{i,j^i} where it is the maximum of all the $V^{i,j}$s. Thus, the output of each subnet represents the maximum WSOPD of all the clusters representing a particular class.

Fig. 1(a) shows K such subnets corresponding to the K classes. These outputs are fed to a winner-take-all neural network (WTANN) [21]. Only one of the outputs corresponding to the maximum will be one, and the rest will be zero. An output of one indicates that the pattern x is assigned to that class.

Let us now look at the training procedure to obtain the desired weights. The first layer corresponds to the extraction of the PC's which form a set of basis vectors. Many neural networks have been designed to extract the PC's [22], [23]. Any of these principal component analysis neural networks (PCANN's) could be used to extract the PC's of each class separately from a given set of training samples. Initially, all the samples belonging to a particular class are used to obtain the PC's corresponding to the first cluster. Subsequently, when a new cluster needs to be introduced, we need to identify those samples corresponding to the maximum entry of the confusion matrix. Identification of the correct class and the wrong class are straight forward. However, identification of the correct and wrong clusters can be achieved by passing the j^i information along with the z^{i,j^i} information from the MWTANN's to the end. Thus, identification of the maximum of the confusion matrix can be done. We can now pass all the patterns once more and identify the required patterns leading to the quadruple corresponding to this maximum. Once the weights of this layer are found out in this manner, they remain fixed until the end.

The w weights are initialized to “1's”. In the case of the GSM, they remain fixed till the end. However, in case of the GLSM, once a new cluster q has been introduced in class p as above, we need to adapt its $w^{p,q}$s. Training patterns are presented one at a time. Let x and ζ be the input and the target output vectors of the current training pattern. If x belongs to class i then ζ_i is one and ζ_j is zero for all $j \neq i$. We pass x through the network and obtain the output y from the WTANN. Then, we compute the reinforcement signal R where $R_i = y_i - \zeta_i, i = 1, \cdots, K$. Note that when the pattern is misclassified, R_i is -1 only for the correct class and R_i is one only for the wrong class. For the other classes R_i is zero. In case of correct classification, all components of R will be zeros. Only R_p is backpropagated to the third layer of the $subsubnet^{p,q}$. The weight changes are computed according to (7) or (8) when R_p is -1 or one, respectively. If R_p is zero and ζ_p is one, then (9) is used to compute the weight changes. Otherwise, the weight changes are assumed to be zeros. This process of training is repeated for a given number of steps.

As mentioned earlier, one of the reasons for the scalability of the growing methods is their possible neural implementation. Here, the extraction of PC's using NN's is the most important step. Using the algebraic techniques, the time and space complexities are $O(n^2)$, where n is the dimensionality. This is because of the fact that correlation matrices of size $n \times n$ have to be dealt with. However, using PCANN's, they are only $O(nN)$, where N is the number of PC's in each cluster and $N \ll n$ for large-dimensional problems. This is because of the ability of these neural networks to extract the first few PC's directly from the data.

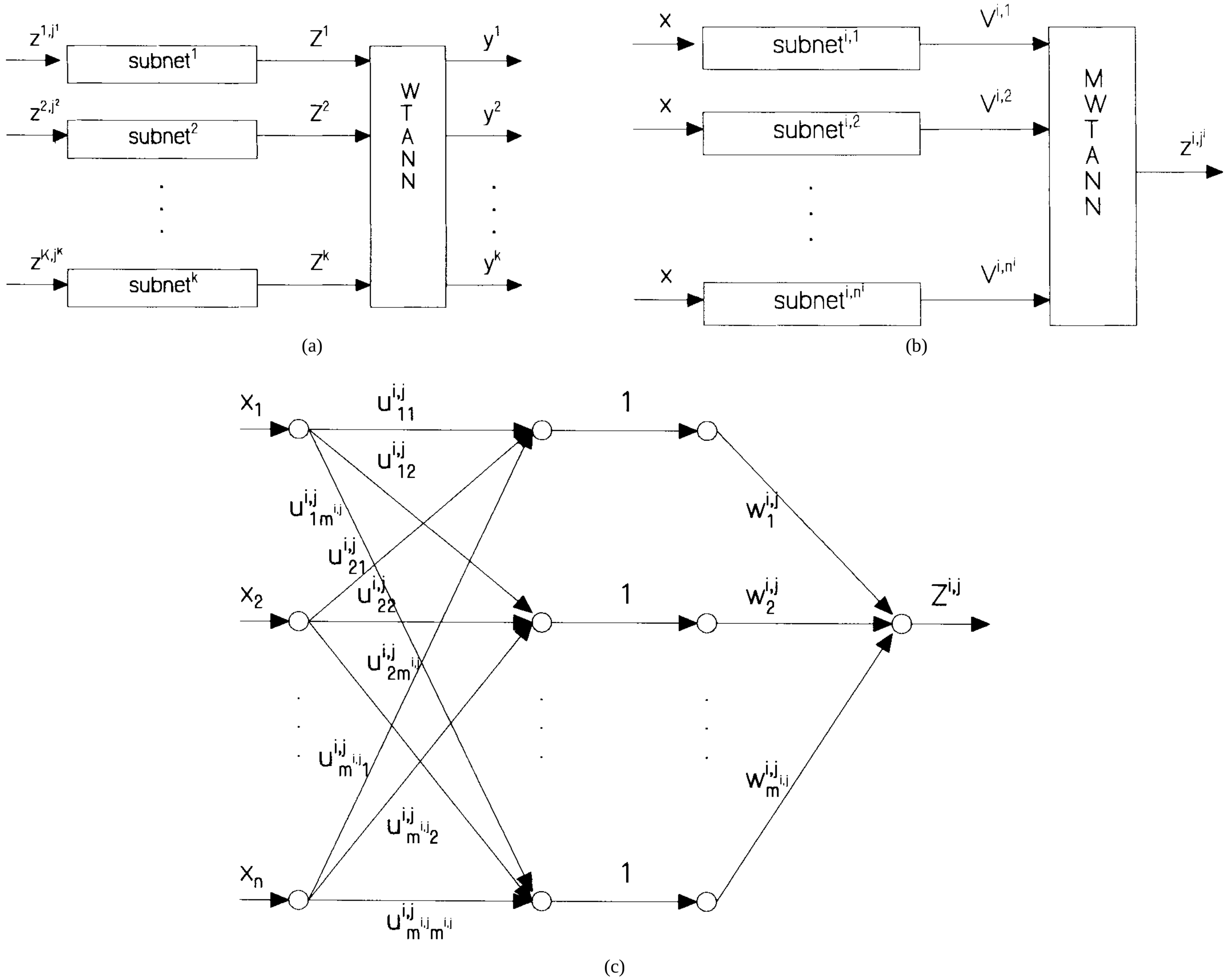

Fig. 1. (a) Block diagram of the neural network. (b) Subnet to compute the maximum WSOPD for all clusters of class i. (c) Subsubnet to compute the WSOPD for the jth cluster of class i.

VI. Experiments and Results

A. Data Sets Used

We have chosen three data sets: the optical character data, the sonar data, and the vowel data.

1) Character Data: This data set consists of handwritten digits, and was used earlier [6], [24], [12]. Each digit is a binary image of size 32 × 24 pixels. There are 13 326 training and 3333 test samples. In [12] the original 13 326 training samples are divided into two near-halves: a training set consisting of 6670 samples, and a validation set consisting of 6656 samples. Further, the dimensionality is reduced by forming nonoverlapping windows of size 2 × 2 over the entire image, and replacing each window by one feature whose value corresponds to the number of bits on in that window. Thus the value of every feature varies from zero to four, and there are a total of 192 features. We have used this data here.

2) Sonar Data: This problem addresses the undersea target identification. A set of 208 sonar returns (one class of 111 cylinder returns and another class of 97 rock returns) were split into a training and a test data set of size 104 each [25].

3) Vowel Data: This problem addresses the identification of 11 steady-state vowels of English spoken by 15 speakers for a speaker normalization study. Extensive results on this data are reported [20]. Both the vowel and the sonar data are electronically available from the connectionist benchmark collection of Carnegie-Mellon University, Pittsburgh, PA [26].

B. Results and Discussion

We have simulated the GSM and the GLSM on all three data sets. We have used the algebraic techniques to extract the PC's instead of the PCANN's. Since we have not taken more than 13 PC's in any of our simulations, the discrepancy in numerical accuracy between both implementations will be negligible. We feel this based on such comparisons made earlier [7].

As mentioned, at the beginning of growing, we will have one cluster per class. This corresponds to the SPRM. In both the GSM and the GLSM, the starting point is this. We have varied the number of PC's per cluster, which is the only important design parameter. The other parameters have been fixed somewhat arbitrarily. MINPATS is one; the number of

TABLE I
CLASSIFICATION RESULTS USING THE SPRM AT THE BEGINNING OF GROWING

SONAR			VOWEL			CHARACTER			
Number	Accuracy (%)		Number	Accuracy (%)		Number	Accuracy (%)		
of PCs	Trg	Test	of PCs	Trg	Test	of PCs	Trg	val	Test
1	64.4	66.3	1	43.0	33.5	1	66.7	66.5	66.4
2	75.0	69.2	2	64.0	43.3	2	74.9	74.7	74.3
3	80.8	81.7	3	70.5	38.7	3	79.3	79.5	79.4
4	76.9	80.8	4	75.2	35.5	9	89.3	87.4	88.4
5	86.5	83.7	5	82.6	37.0	11	90.4	88.7	89.0
6	85.6	81.7	6	84.3	29.2	13	91.2	88.7	88.9

TABLE II
CLASSIFICATION RESULTS USING THE GSM AND THE GLSM ON THE CHARACTER DATA

GSM						GLSM					
Number	Accuracy (%)			Total number of		Number	Accuracy (%)			Total number of	
of PCs	Trg	val	Test	clusters	PCs	of PCs	Trg	val	Test	clusters	PCs
1	89.5	84.6	85.1	192	192	1	89.1	82.6	83.6	192	192
2	94.9	86.1	86.0	186	372	2	94.5	86.0	85.4	191	382
3	97.9	87.4	87.9	184	552	3	97.7	88.1	88.1	181	543
9	100.0	90.1	90.8	94	846	9	100.0	89.7	91.0	108	972
11	100.0	90.5	90.8	77	847	11	100.0	90.4	90.7	85	935
13	100.0	90.8	90.8	49	637	13	100.0	90.6	90.8	77	1001

iterations of weight adaptations, HLSMITERS, is 200, 500, and 500 for the character, the sonar, and the vowel data sets, respectively; η is zero in all cases; η^+ $(= \eta^-)$ is 1.0e-9, 1.0e-6, and 1.0e-6 for the character, the sonar, and the vowel data sets, respectively; MAXCLUSTERS is 20 in all cases.

The results of the SPRM for all data sets are presented in Table I. The results using the GSM and the GLSM are presented separately for each data set in Tables II–IV. For a better visualization, we present results for the GSM and the GLSM in Fig. 2. Fig. 3 shows how the classification accuracy varies as clusters are introduced one by one in the GSM. For this illustration, we have chosen the results corresponding to only the best case from each data set. Best case here corresponds to the maximum classification accuracy on the validation/test set. Based on these results (i.e., Tables I–IV and Figs. 2 and 3), the following observations can be made.

1) On all data sets, both the GSM and the GLSM could achieve 100% accuracy on the training data as can be seen from Fig. 3 and Tables II–IV. Hence, we can conclude that both the growing methods have the potential to approximate any decision surface. Thus, one of the motivations has been achieved.
2) On the training sets, the classification accuracy increases with an increase in the number of PC's per cluster for all data sets as can be seen from Tables II–IV. On the validation/test sets, as the number of PC's per cluster increases, the classification accuracy increases initially, reaches a peak, and starts decreasing as can be seen from Table II and Fig. 2. This peaking phenomenon occurs because of overfitting that arises out of an increase in the number of free parameters resulting out of an increase in PC's. However, tuning of the design parameter is very simple.
3) From Fig. 2, we observe that the GLSM could improve upon the GSM only marginally in terms of classification accuracy.
4) The average time taken for convergence of the GSM and the GLSM on the character data is approximately 15 min and 5 h, respectively, on a Pentium-based machine. This is the total execution time, and includes a lot of input–output. Hence, the GSM has a much better potential at scalability than the GLSM.

In order to be able to compare the proposed algorithms with the other methods including the EALSM, we have summarized the relevant results in Tables V and VI. Based on them, we can infer the following.

Classification Accuracy: The growing methods are slightly inferior to the EALSM on the sonar and the vowel data. However, on the character data, they are as good as the EALSM. The growing methods are better than the ALSM on the sonar and vowel data and are comparable to it on the character data. Also, they are comparable to the other methods for both the character and the sonar data, although they are inferior for the vowel data.

TABLE III
CLASSIFICATION RESULTS USING THE GSM AND THE GLSM ON THE SONAR DATA

GSM					GLSM				
Number	Accuracy (%)		Total number of		Number	Accuracy (%)		Total number of	
of PCs	Trg	Test	clusters	PCs	of PCs	Trg	Test	clusters	PCs
1	100.0	83.7	28	28	1	100.0	78.8	38	38
2	100.0	86.5	19	38	2	100.0	86.5	15	30
3	100.0	87.5	11	33	3	100.0	84.6	8	24
4	100.0	86.5	8	32	4	100.0	89.4	8	32
5	100.0	85.6	5	25	5	100.0	85.6	4	20
6	100.0	82.7	7	42	6	100.0	80.8	5	30

TABLE IV
CLASSIFICATION RESULTS USING THE GSM AND THE GLSM ON THE VOWEL DATA

GSM					GLSM				
Number	Accuracy (%)		Total number of		Number	Accuracy (%)		Total number of	
of PCs	Trg	Test	clusters	PCs	of PCs	Trg	Test	clusters	PCs
1	98.3	38.5	138	138	1	70.8	35.1	199	199
2	100.0	40.0	92	184	2	94.1	43.1	118	236
3	100.0	34.6	63	189	3	99.8	34.6	68	204
4	100.0	33.1	40	160	4	98.7	33.3	68	272
5	100.0	27.7	35	175	5	99.6	31.8	64	320
6	100.0	25.1	47	282	6	100.0	26.8	34	204

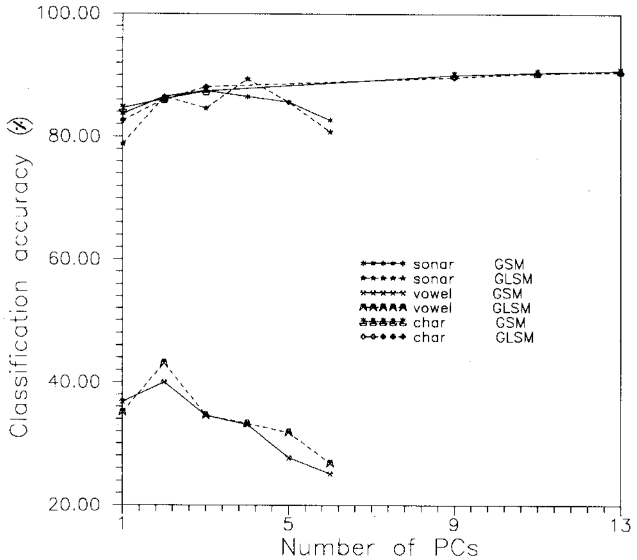

Fig. 2. Number of PC's versus classification accuracy for the GSM and the GLSM for the validation/test sets.

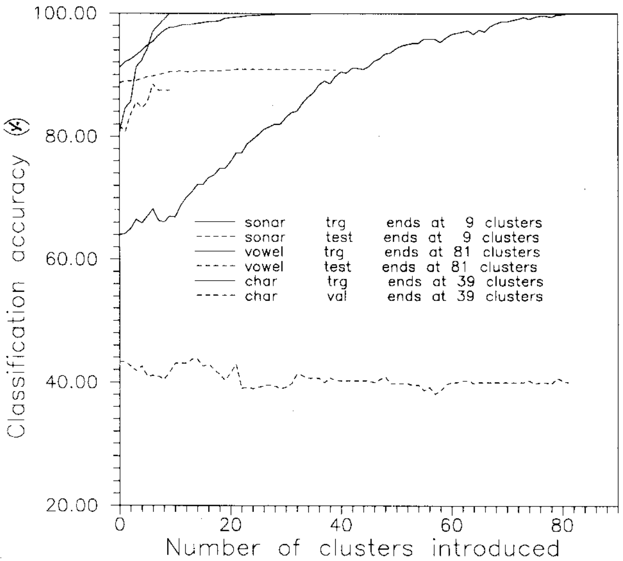

Fig. 3. Number of clusters introduced versus classification accuracy for the GSM.

Classification Speed: All subspace methods are comparable as only dot products are involved. They will be better than the other methods as nonlinearity computation in the MLP's, while the RBF's and the GCS will turn out to be costlier. In case of the character data, the feature extraction is extremely time consuming. In case of the NNC, one will have to compare with all the 13 326 (their) training patterns. Thus, on large-dimensional problems, our methods can be much faster.

Ease of Design: From the point of view of determining the design parameters, the growing methods are superior to the EALSM as they have eliminated the necessity to determine the number of clusters in each class by trial-and-error. Since

TABLE V
COMPARISON OF RESULTS OBTAINED BY DIFFERENT SUBSPACE METHODS

Data set	Classification	Accuracy (%)			Total	Reported in
	method	Trg	val	Test	PCs	Reference number
CHAR	EALSM	100.0	91.9	92.0	99	[27]
	SPRM	91.2	88.7	89.0	130	This paper
	ALSM	99.1	90.5	91.1	140	This paper
	GSM	100.0	90.8	90.8	847	This paper
	GLSM	100.0	90.6	90.8	1001	This paper
SONAR	EALSM	100.0	-	93.3	20	[27]
	SPRM	86.5	-	83.7	10	This paper
	ALSM	100.0	-	85.6	12	This paper
	GSM	100.0	-	87.5	33	This paper
	GLSM	100.0	-	89.4	32	This paper
VOWEL	EALSM	100.0	-	45.9	99	[27]
	SPRM	64.0	-	43.3	184	This paper
	ALSM	76.1	-	37.2	33	This paper
	GSM	100.0	-	40.0	184	This paper
	GLSM	98.5	-	41.6	380	This paper

TABLE VI
OTHER KNOWN RESULTS FOR THE SELECTED DATA SETS

Data set	Classifier	Features	Accuracy (%)	Reported in
		or hidden units	Test	Reference number
CHAR	NNC	40 Zernike	86.4	[24]
	NNC	100 topological and statistical	90.5	[6]
	MLP	100 topological and statistical	90.4	[6]
SONAR	MLP	12 hidden units	90.4	[25]
VOWEL	MLP	88 hidden units	51.0	[20]
	RBF	528 hidden units	53.0	[20]
	GCS	154 hidden units	67.0	[20]

these methods exhibit a peaking behavior, tuning the design parameters is easy.

Scalability: From both time and space complexities, the growing methods are better than the EALSM, especially on large-dimensional problems. The time taken on the character data by the GSM, the GLSM, and the EALSM is 15 min, 5 h, and 2.5 h, [27] respectively, on the same machine. If simulations to extract PC's is based on neural-network implementations, then the time will be even less for the growing methods.

VII. SUMMARY AND CONCLUSION

In this paper, we have proposed two growing subspace algorithms. The design of these algorithms is inspired by the constructive neural-network architectures. They could successfully eliminate two problems associated with the EALSM. By introducing one cluster at a time and adapting it if necessary, we could successfully eliminate the difficult issue of deciding how many clusters to have in each class. The degradation in classification accuracy is small. The growing methods, like the EALSM, have the potential to approximate any decision surface. The limitation of scalability of the EALSM could be overcome by these methods by the above strategy in addition to the neural-network implementation. We have conducted experiments on three data sets including a 192 dimensional large character data. We have shown that these methods are comparable to other classifiers like the MLP and the NNC in terms of classification accuracy. In terms of classification speed and ease and scalability of design, they are better for large-dimensional problems. In addition to these, the appropriate features are extracted automatically in these methods. Thus, we feel that the subspace methods have the potential to become general purpose classifiers for large-dimensional problems where magnitude information is not crucial for discrimination.

References

[1] S. Raudys and A. K. Jain, "Small sample size effects in statistical pattern recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, pp. 252–264, 1991.

[2] A. K. Jain and R. Chandrasekaran, "Dimensionality and sample size considerations in pattern recognition practice," in *Handbook of Statistics*, vol. 2, *Classification, Pattern Recognition, and Reduction of Dimensionality*, P. R. Krishnaiah and L. N. Kanal, Eds. New York: North-Holland, 1982.

[3] A. K. Jain and K. Karu, "Learning texture discrimination masks," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, pp. 195–205, 1996.

[4] J. Mao and A. K. Jain, "Artificial neural networks for feature extraction and multivariate data projection," *IEEE Trans. Neural Networks*, vol. 6, pp. 296–317, 1995.

[5] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropogation applied to handwritten zip code recognition," *Neural Computa.*, vol. 1, pp. 541–551, 1989.

[6] W. E. Weideman, M. T. Manry, and H. C. Yau, "A comparison of a nearest neighbor classifier and a neural network for numeric handprint character recognition," in *Proc. Int. Joint Conf. Neural Networks*, Washington, D.C., 1989, pp. 117–120.

[7] L. Todd, R. Mike, and H. Dar, "Hebbian feature discovery improves classifier efficiency," in *Proc. Int. Joint Conf. Neural Networks*, San Diego, CA, 1990, pp. I-51–I-56.

[8] J. Mao, K. Mohiuddin, and A. K. Jain, "Parsimonious network design and feature selection through node pruning," in *Proc. ICPR*, Jerusalem, Israel, 1994, pp. 622–624.

[9] A. K. Jain and J. Mao, "Neural networks and pattern recognition," in *Computational Intelligence for Imitating Life*, J. Zuruda, R. J. Marks, and A. J. Robinson, Eds. Piscataway, NJ: IEEE Press, 1994, pp. 194–212.

[10] S. Watanabe and N. Pakvasa, "Subspace method in pattern recognition," in *Proc. Int. Joint Conf. Pattern Recognition*, 1973, pp. 25–32.

[11] E. Oja, *Subspace Methods of Pattern Recognition*. Letchworth, U.K.: Res. Studies Press, 1983.

[12] M. Prakash and M. N. Murty, "Hebbian learning subspace method: A new approach," *Pattern Recognition*, 1995, in press.

[13] T. Kohonen, G. Nemeth, K. Bry, M. Jalanko, and K. Makisara, "Spectral classification of phonemes by learning subspace methods," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Washington, D.C., 1979, pp. 97–100.

[14] L. Xu, A. Krzyzak, and E. Oja, "Neural-net method for dual subspace pattern recognition," in *Proc. Int. Joint Conf. Neural Networks*, Seattle, WA, 1991, pp. II-379–II-384.

[15] E. Oja and T. Kohonen, "The subspace learning algorithm as a formalism for pattern recognition and neural networks," in *Proc. 1988 IEEE ICNN*, San Diego, CA, pp. 277–284.

[16] E. Oja and M. Kuusela, "The ALSM algorithm—An improved subspace method of classification," *Pattern Recognition*, vol. 16, no. 4, pp. 421–427, 1983.

[17] E. Oja and J. Parkkinen, "Texture subspaces," in *Pattern Recognition Theory and Applications*, P. A. Devijver and J. Kittler, Eds. New York: Springer-Verlag, 1986, pp. 21–22.

[18] M. Prakash and M. N. Murty, "Extended subspace methods of pattern recognition," *Pattern Recognition Lett.*, 1995, in press.

[19] S. E. Fahlman and C. Lebiere, "The cascade-correlation learning architecture," in *Advances in Neural Information Processing*, D. S. Touretzky, Ed. San Mateo, CA: Morgan Kaufmann, 1989, pp. 524–532.

[20] B. Fritzke, "Growing cell structures—a self-organizing network for unsupervise and supervised learning," Int. Comput. Sci. Inst., Berkeley, CA, Tech. Rep. TR-93-026, 1993.

[21] J. Hertz, A. Krogh, and R. G. Palmer, *Introduction to the Theory of Neural Computation*. Reading, MA: Addison-Wesley, 1991.

[22] E. Oja, "Neural networks, principal components and subspaces," *Int. J. Neural Syst.*, vol. 1, pp. 61–68, 1989.

[23] T. D. Sanger, "An optimality principle for unsupervised learning," in *Advances in Neural Information Processing*, D. S. Touretzky, Ed., pp. 11–19, 1989.

[24] K. Alireza and Y. H. Hong, "Rotation invariant image recognition using features selected via a systematic method," *Pattern Recognition*, vol. 23, no. 10, pp. 1089–1101, 1990.

[25] R. P. Gorman and J. Sejnowski, "Learned classification of sonar targets using a massively parallel network," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-36, pp. 1135–1140, 1988.

[26] S. E. Fahlman, *CMU Benchmark Collection for Neural-Net Learning Algorithms*, Carnegie-Mellon Univ., School Comput. Sci., Pittsburg, PA, 1993.

[27] M. Prakash, "Learning in subspace methods using neural network and genetic approaches," Ph.D. dissertation, Indian Inst. Sci., Bangalore, under preparation.