A Composite Classifier System Design: Concepts and Methodology

BELUR V. DASARATHY, SENIOR MEMBER, IEEE, AND BELUR V. SHEELA

Abstract—This study explores the scope for achieving enhanced recognition system performance through deployment of a composite classifier system consisting of two or more component classifiers which belong to different categories. The domains of deployment of these individual components (classifiers) are determined by optimal partitioning of the problem space. The criterion for such optimal partitioning is determined in each case by the characteristics of the classifier components. An example, in terms of partitioning the feature space for optimal deployment of a composite system consisting of the linear and nearest neighbor (NN) classifiers as its components, is presented to illustrate the concepts, the associated methodology, and the possible benefits one could expect through such composite classifier system design. Here, the optimality of the partitioning is dictated by the linear class separability limitation of the linear classifier and the computational demand characteristics of the NN classifier. Accordingly, the criterion for the optimal feature space partitioning is set to be the minimization of the domain of application of the NN classifier, subject to the constraint that the linear classifier is to be deployed only in regions satisfying the underlying assumption of linear separability of classes. While many alternatives are available for the solution of the resulting constrained optimization problem, a specific technique—Sequential Weight Increasing Factor Technique (SWIFT)—was employed here for convenience in view of previous successful experience with this technique in other application areas. Numerical results derived using the well-known IRIS data set are furnished to demonstrate the effectiveness of the new concepts and methodology.

I. Introduction

PATTERN-recognition system design, be it statistical or syntactic in its concept, has generally been viewed as a problem in identifying and determining the parameters characterizing a specific type of classifier which is best suited to the problem environment. One could, however, visualize problem environments wherein a single type of classifier may not necessarily represent the best choice over the entire problem space. This calls for an exploration of the concept of composite classifier systems. Recognition of certain pattern classes may demand a syntactic approach while others may be best discriminated through a statistical pattern classification technique. This requires partitioning of the problem space to define the domains of deployment of the statistical and syntactic components of the composite system. Partitioning of the problem space at this level is essentially on a heuristic basis at the present time, and current efforts have been directed mostly towards the development of a hierarchical approach involving use of the syntactic and statistical approaches at different problem levels as appropriate [1]. The present study, therefore, restricts itself to the realm of statistical pattern recognition techniques pending development of additional concepts necessary to formulate an optimal approach to the partitioning of the problem space at the highest level of using both syntactic and statistical techniques. The problem of partitioning the problem space for deploying the components of a composite classifier system, within the realm of statistical techniques, can be viewed as one of partitioning the feature space without loss of generality, provided the feature space encompasses all the features considered significant for discrimination of each and every pattern class in the problem environment. This is pertinent in view of the fact that selection of different feature subsets for discrimination of different pattern classes in a single problem environment has been found to lead to improved performance in multiclass environments [2]. Depending on the shape and complexity of cluster separability, different pairs of pattern classes may, for best discrimination, require not only different feature subsets but also different types of classifiers employing these different feature subsets. Even for a given pair of pattern classes, a single type of classifier may not prove to be the ideal choice over the entire feature subspace spanned by this pair. An intuitively appealing concept is to permit deployment of a composite system with two or more classifier components (each of which is of a different category) by partitioning the feature space and defining individual subspaces over which each classifier component is to be employed. Such partitioning should be based on some rationale which will tend to enhance the overall recognition system performance. In other words, the partitioning should be optimal with respect to some optimality criterion defined externally to accomplish enhanced system performance. An obvious performance characteristic desired of a recognition system is maximum computational economy with minimum loss in recognition accuracy. However, the specific structure of this optimality criterion is dependent on the characteristics of the classifier components of the composite system. For illustrative purposes, we shall now consider in the following sections a composite system consisting of linear and nearest neighbor classifiers as its components and develop the optimality criterion for partitioning the feature space in the context of this type of composite system. This will be followed by details of the composite system design methodology and structure as well as simulation experience.

Manuscript received March 31, 1978; revised July 14, 1978. A preliminary version of this paper was presented at the Pattern Recognition and Image Processing Conference, Chicago, IL, May 1978.
B. V. Dasarathy is with M&S Computing, Inc., Huntsville, AL 35805.
B. V. Sheela is with the ISRO Satellite Center, Bangalore, India.

II. A Composite Linear/Nearest Neighbor Classifier System Design

It is well known that, among all nonparametric and parametric classification techniques, the linear classifier makes one of the lowest computational demands as it generally requires only m (the number of pattern classes) n-dimensional

0018-9219/79/0500-0708$00.75 © 1979 IEEE

vector multiplications for one test sample classification. However, its performance/recognition accuracy is subject to the validity of the assumption that the pattern classes are in fact linearly separable. But, often, this assumption is not strictly valid, as the so-called optimal hyperplane derived under this assumption will still misrecognize at least a small percentage of the training sample set itself. Thus while the linear classifier satisfies very well one of the two prescribed objectives, viz minimum computational complexity, the other objective may not be satisfied fully depending on the true extent of cluster separability. This, therefore, represents a potential domain for composite classifier system design and deployment, provided one can choose the appropriate type of classifier for the second component to complement the (first) linear classifier. The choice of the type of this second classifier component will in turn dictate the formulation of the optimality criterion for partitioning the feature space between the two chosen classifier subsystems. Thus in essence one has to determine the region of conflict, i.e., the region wherein linear separability is not valid, and identify a more "sophisticated" classifier for deployment within this region of conflict. The choice of this second classifier is of course open and depends on the complexity of the separability of classes in this subspace. Assuming that this is in the nonparametric realm (because determining the probabilistic descriptions of the data within the region of conflict may prove to be difficult in view of the relatively small number of samples in this region), a candidate worthy of serious consideration is the nearest neighbor (NN) technique or one of its many modifications [3]. The advantages of using NN techniques are obvious. They require neither a probabilistic description of the distributions underlying the pattern classes nor a knowledge of the functional form of the nonparametric discriminants separating the classes. This admits discrimination in complex cluster environments. Further, the recognition errors have known upper bounds [4]. Their only, but significant, disadvantage is the relatively high computational demands which, in spite of the extensive research in the area, are considerably more than that of other classifiers [3]. But deployment of NN techniques in this context along with the linear classifier is an attractive proposition, as the number of samples in the region of conflict, and, hence, the number of distance computations, is small. Thus an ideal composite classifier system would be one that can combine the conceptual flexibility and bounded recognition error environment of the NN techniques with the computational economy (and perfect recognition outside the regions of conflict) of the linear classifier. This indeed is possible provided one can define the domains of applicability of the two classifiers in such a way as to retain their individual advantages while compensating out their disadvantages.

We, therefore, now need to develop the optimality criterion in specific terms having identified through qualitative judgment the two components of the composite classifier system. As the aim is to retain the computational economy of the linear classifier to the extent feasible, it is advantageous to employ a linear classifier as far as possible and limit the use of NN classifier only to cases wherein the decision of the linear classifier is in doubt. This means we have to find a minimal region of conflict (a union of subspaces bounded by a pair of hyperplanes for each pair of pattern classes). In other words, the optimization criterion effectively becomes minimization of the number of samples (of the given training sample set) in a region of conflict defined by the linear classifier, and the optimization problem is to define this region of conflict (through pairs of hyperplanes) such that the number of samples therein is minimum. It should be noted that the samples in the region of conflict include not only samples classified wrongly by the linear classifier but also those whose classification would be in doubt if their classification labels were to be unknown *a priori*.

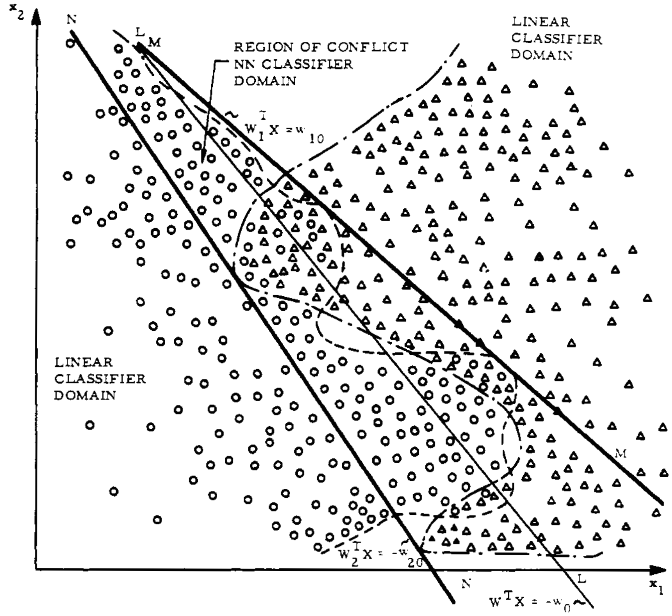

Fig. 1. An illustrative example of a two-class problem.

This approach leads to a system which employs the linear classifier in the region of assured recognition (100 percent for the training sample set and close to 100 percent for the test data depending on how closely representative of the test set the training set really is) with minimum computational demands most of the time and the NN classifier for only those samples falling within the region of conflict wherein the recognition error has an upper bound (twice the Bayes error). Here the computational demand, while greater than in the other region, is still minimized as the number of neighbor distances to be computed has been minimized. This conceptual design, therefore, calls for the solution of the optimization problem of minimizing the number of samples in a region of conflict defined by the linear classifier. In the following sections, we shall show that this is effectively a constrained optimization problem, present a method of approach for solving this constrained optimization problem, and share the results of our simulation experience which brings out the value of this newly proposed concept.

III. The Constrained Optimization Problem

Consider for illustrative purposes a two-class problem environment defined over a two-dimensional feature space as shown in Fig. 1. Therein, L-L represents the so-called optimal linear classifier hyperplane whose equation is given by $w_0 + W^T X = 0$ where $W^T = [w_1, w_2]$ and X is the feature vector: $X = [x_1, x_2]$. It is clear that this optimal classifier is optimal (depending on the method used in deriving the hyperplane [5]-[9]) only in the sense that it represents the best classification under the assumption of linear separability and perhaps minimizes the number of errors [7]. While this assumption is invalid in the example shown, the linear classifier is still fully satisfactory in the feature space outside the region of overlap/conflict bounded by the discriminants N-N and M-M.

In the ideal case when the two classes are linearly separable,

$$w_0 + W^T X \geqslant 0, \quad \forall X \in C_1$$
$$< 0, \quad \forall X \in C_2. \tag{1}$$

However, when such is not the case, then there will exist a subset χ_s of χ (the training sample set) for which these inequalities are violated. Under these conditions, one can define a pair of hyperplanes N-N and M-M such that

$$W_2^T X + w_{20} > 0, \quad \forall X \in C_1 \tag{2}$$
$$W_1^T X - w_{10} < 0, \quad \forall X \in C_2. \tag{3}$$

This defines a region of conflict given by

$$\{(W_2^T X > -w_{20}) \cap (W_1^T X < w_{10})\}$$

where

$$X \in \chi_s (C_1 \text{ or } C_2).$$

Now the problem is to find the w's such that the number of samples in χ_s is minimum subject to the constraints (2) and (3). Equivalently, we can define the constrained optimization problem as the following:

given

$$\chi = [X_1, X_2, \cdots, X_p], \qquad \mathcal{L} = [L_1, L_2, \cdots, L_p]$$

where

$$X_i = [x_i^1, x_i^2, \cdots, x_i^n]^T \qquad L_i = k | X_i \in C_k, \quad k = 1, 2.$$

Determine

$$W_j = [w_j^1, w_j^2, \cdots, w_j^n]^T \text{ and } \delta_j; \ j = 1, 2$$

such that

$$J(W_j, \delta_j | j = 1, 2) \triangleq \sum_{X_i \in \chi} u_1^i u_2^i \text{ is minimum} \tag{4}$$

under the inequality constraints

$$d_2^i > -\delta_2, \quad \forall X_i \in C_1$$
$$d_1^i < \delta_1, \quad \forall X_i \in C_2 \tag{5}$$

where

$$u_1^i = [|d_1^i - \delta_1| - (d_1^i - \delta_1)]/2|d_1^i - \delta_1|$$
$$u_2^i = [|d_2^i + \delta_2| + (d_2^i + \delta_2)]/2|d_2^i + \delta_2|$$
$$d_j^i = X_i^T W_j.$$

Here

$$u_1^i \cdot u_2^i = 1, \quad \forall X_i | d_1^i < \delta_1 \text{ and } d_2^i > -\delta_2, \quad \text{i.e., } \forall X_i \in \chi_s$$
$$u_1^i \cdot u_2^i = 0, \quad \forall X_i | d_1^i \geqslant \delta_1 \text{ or } d_2^i \leqslant -\delta_2, \quad \text{i.e., } \forall X_i \notin \chi_s.$$

While the constrained optimization problem stated above can be solved by any one of the approaches available in this field for nonlinear programming problems, the penalty function approach was chosen here [10]. Admittedly, these methods do not guarantee total constraint satisfaction as they, in effect, offer a tradeoff between the minimization of objective function and the satisfaction of constraints. The training sample set being only representative of the test samples to be classified in the operational phase, exactly 100 percent recognition of this set, while desirable, is not very essential for achieving high recognition efficiency in the operational phase. Under this minimal conceptual compromise, however, we gain significantly on the computational side in being able to treat the problem as a sequence of equivalent unconstrained optimization problems derived by adding weighted functions of the constraints to the objective function. This weighting factor in most methods is chosen arbitrarily and imposed extrinsically on the problem. Recently, a new method—Sequential Weight Increasing Factor Technique (SWIFT)—was developed which internally determines the weightage of the penalty term at each iteration on the basis of the results of the previous iteration [10], [11]. Further, the method does not necessarily need to be started in a region with guaranteed constraint satisfaction. Accordingly, this method is adapted here for the solution of this constrained optimization problem of partitioning the feature space. However, the details of SWIFT, being available elsewhere in the literature, are omitted here [10], [11]. It should be noted here that the main purpose of this study is the exploration and development of the concept of composite classifier system design and not the development or presentation of the best constrained optimization technique. As such, no attempt is made to carry out any relative assessment to identify the best possible tool for this subproblem. SWIFT does satisfy this limited purpose, and, in view of the successful experience with this method elsewhere in other applications, it has been adapted here. However, one is always free to adapt his (or her) own pet constrained optimization (algorithm) tool into the structure developed here.

IV. The Recognition System

As shown in Fig. 2, the new recognition system has basically the same structure as the classical system and is composed of the feature selector, training or classifier-design subsystem, and the classification subsystem. The detailed structures of the subsystems are, however, significantly different. The classifier, for example, consists of two components (instead of one): the linear classifier pair and the NN classifier. Correspondingly, the classifier design has two parts. The feature selector permits selection of nonidentical feature subsets for the different classifiers as well as for different pattern classes.

We should note in this context the following points about the computational efforts vis-à-vis the error rate involved under the new system in order to be able to appreciate the enhanced performance achieved by the composite system concept.

The computational effort of consequence to an operational classification system is mainly that expended by the classifier subsystem in classifying the incoming unknown samples. The computational effort in learning is only of secondary importance, and an extra effort in this phase is well worth the expense if it can lead to a savings in the operational phase. This is the rationale for expending the extra computational effort of solving the optimization problem in order to gain the computational savings of not having to use the NN classifier over the whole domain. In most cases, i.e., whenever the unknown sample falls outside the region of conflict, the computational effort is comparable to that under the

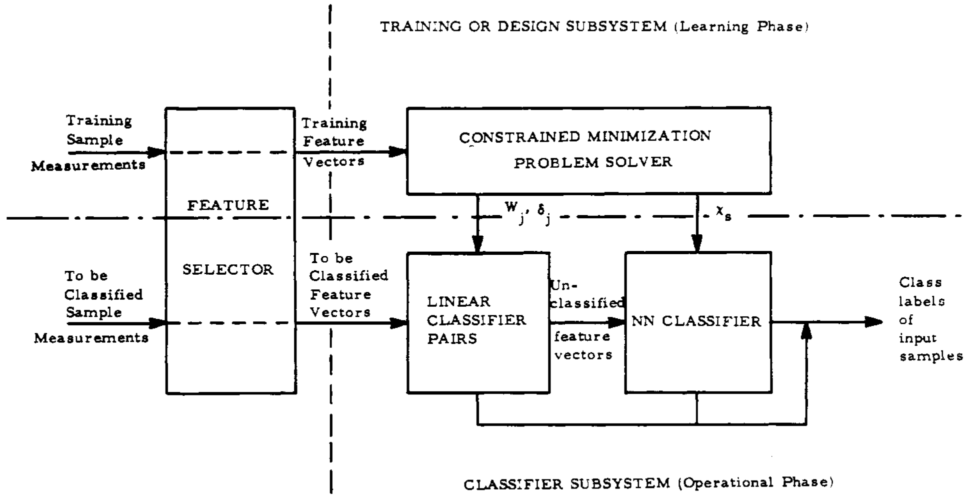

Fig. 2. Schematic representation of the new recognition system based on the integrated linear/NN classifier design.

TABLE I
RELATIVE MERITS AND DEMERITS OF THE LINEAR, NN, AND COMPOSITE CLASSIFIERS

Property	Linear Classifier	NN Classifier	Composite Classifier
Domain of Application	Limited to Linearly Separable Classes (Error bounds unknown in other cases)	Not Limited to Linearly Separable Classes	Not Limited to Linearly Separable Classes
Classification Error in a General Case	0% outside region of conflict but unbounded within this region and hence overall bound unknown in a general case.	Bounded by twice the Bayes error in the entire region.	0% outside the region of conflict and bounded by twice the Bayes error within this region.
Computational Effort			
a. Learning Phase	Low	Negligible*	Medium
b. Classification Phase	Low	Very High	Low-Medium
Merit Summary	Minimum computational effort	Applicable to a general case with known upper bound on errors.	Low computational efforts with known upper bounds on errors and applicable to a general case.
Demerit Summary	Limited to linearly separable classes; otherwise, errors are unbounded.	Maximum computational demands	Unsuited for applications with very few test (to be classified) samples as design/learning phase needs extra effort.

*Provided no training sample set size reduction is carried out.

linear classifier (the sample is checked against a pair of hyperplanes instead of only one). In the small number of cases wherein the unknown sample falls within the region of conflict, the computational effort is moderately higher and involves an additional p distance computations (where p is the number of training samples found in the region of conflict on solving the constrained optimization problem). However, even this effort, expended in a minority of cases only, is still far less compared to the number of distances to be computed for each sample classification under the classical use of NN rule (i.e., of using only the NN classifier). This is true irrespective of the type of NN rule used alone as well as in the composite system.

Thus the structure of the pattern classification system envisaged here saves on computational effort in the classification phase (of course with some additional expense in the learning phase—one cannot get something for nothing) relative to the NN classifier in two distinct ways: 1) by reduction of the number of cases wherein the NN approach is to be used, and 2) by reduction of the number of distances to be computed even in cases where the NN approach is used. Further, in the large number of cases where the NN approach is not at all used, it ensures 100-percent recognition. In those cases where the NN classifier is called for, it retains the bounded error status.

On the other hand, relative to the linear classifier, the composite system, although on the average slightly more expensive computationally, is able to operate with bounded errors even in cases where linear separability is not valid. These facts are presented in summary form in Table I. Thus the

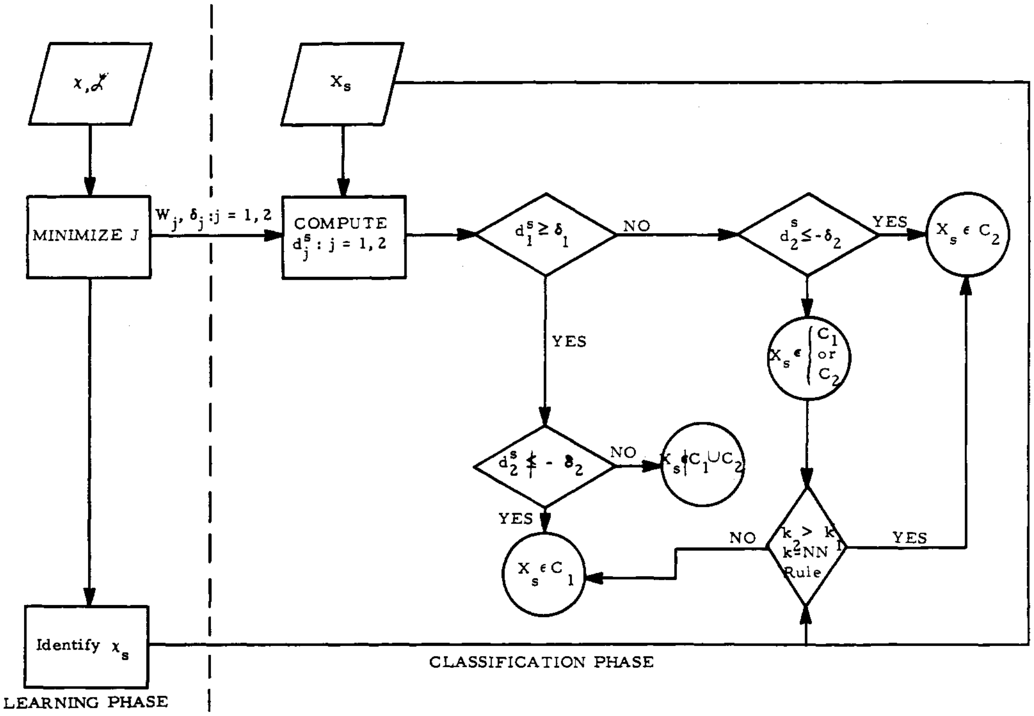

Fig. 3. Flowchart representation of the integrated classification system.

recognition system developed here brings together the linear and NN classifiers in such a way as to combine their advantages while compensating out their disadvantages.

The algorithmic steps involved in the learning and operational phases are furnished below to bring out the conceptual steps as well as the computational effort involved under the new system. These are also presented in the form of a flowchart in Fig. 3 to facilitate an appreciation of the information flow in the system.

A. The Learning Algorithm

Step 1: Input the training sample set: χ.

Step 2: Define a pair of hyperplanes for initiation of the iterative optimization procedure. A suggested approach is to find the best linear discriminant hyperplane for separating the classes using one of the known approaches [5]-[9] and use this as the starting position for both the hyperplanes.

Step 3: This consists of all the steps involved in SWIFT [10], [11] repeated until the appropriate convergence criteria is met leading to determination of optimal values for W_j and δ_j.

Step 4: Identify the subset χ_s (of χ) for which $d_1^s < \delta_1$ and $d_2^s > -\delta_2$.

Step 5: Output to the classifier χ_s, W_j, and δ_j.

B. The Classification Algorithm

Step 1: Input sample X_s to be classified.

Step 2: Compute d_j^s: $j = 1, 2$.

Step 3: $d_1^s \geq \delta_1$ and $d_2^s \nleq -\delta_2 \Rightarrow X_s \in C_1$; go to Step 5. $d_2^s \leq -\delta_2$ and $d_1^s \ngeq \delta_1 \Rightarrow X_s \in C_2$; go to Step 5. $d_1^s < \delta_1$ and $d_2^s > -\delta_2 \Rightarrow X_s \in C_1$ or C_2 (the region of conflict). Go to Step 4. ($d_1^s \geq \delta_1$ and $d_2^s \leq -\delta_2 \Rightarrow X_i \notin (C_1 \cup C_2)$, i.e., the sample is quite unlike the training sample set, and the labeling of the sample is at best arbitrary.)

TABLE II
TYPICAL SOLUTIONS TO THE IRIS DATA SET RELATED CONSTRAINED OPTIMIZATION PROBLEM

Run No.	No. of Samples In Region of Conflict	Constraint Violation Measure: No. of Samples In Violation	Constraint Violation Measure: Summed Extent of Violation	Remarks
1	0	3	0.860621	Initiation Point - Generated by Ho-Kashyap Algorithm
	1	2	0.624712	
	0	3	0.612568	Typical Intermediate Solutions
	4	1	0.000003	
	4	2	0.000002	Final 'optimal' position
2	48	2	0.328051	Arbitrary Initiation point
	14	2	0.048323	
	16	1	0.025619	Typical Intermediate results
	20	0	0.0	
	19	1	0.000165	Final 'optimal' position
3	0	2	0.077697	Initiation point - Generated by an optimal partitioning algorithm for discriminant learning. Final position same - No Improvement

Step 4: Classify X_s using a k-NN rule with χ_s as the prototype set.

Step 5: Output label L_s of X_s, increments $s = s + 1$, and return to process the next sample and repeat Step 1 through Step 5 till all the samples are classified.

V. THE SIMULATION RESULTS

The proposed system was simulated and exercised on a PDP-11/70 using the two-class IRIS data set consisting of 100 four-dimensional samples. Table II presents the results of the iterative optimization procedure. The tradeoff between constraint satisfaction and minimization of the objective function is brought out by presenting typical levels of constraint violations and corresponding minimization values of number of samples in the region of conflict during the progress of the iterative process over several runs. For example, in run 2, the constraint violation reached zero when the number of samples in χ_s was 20. Further minimization resulted in a single constraint violation of a negligible level with χ_s having 19 samples. If a higher level of constraint violation were to be tolerated for this single sample, χ_s could have even lesser samples (16). Thus this represents a tradeoff between the accuracy derived of the linear classifier versus computational demands in the NN classifier. Increasing the domain of the NN classifier increases both the need for using the NN classifier (by increasing the probability of a test sample falling in this domain) as well as the number of neighbor distances to be computed for each sample classification. On the other hand, increasing the domain of the linear classifier decreases the computational demands while correspondingly reducing the 100-percent recognition accuracy achievable in theory with no constraint violation. The resolution of the tradeoff is thus dependent on the application and its priorities in terms of the recognition accuracy and computational economy. (In the limit, the recognition accuracy of the linear classifier, if sufficient in itself, when deployed over the whole feature space precludes altogether the need for this mode of composite classifier design.) Now, looking at run 1, it is seen that the performance is considerably better (in that χ_s has far fewer samples) as the initiation point of the search in that case was not arbitrary but generated as the "best" linear discriminant by the Ho–Kashyap algorithm [6]. An even more "optimal" initiation point (run 3), in the sense of having lesser constraint violation, obtained using a recently developed optimal partitioning algorithm for linear discriminant learning, did not, however, lead to any improved performance [12]. This shows a certain amount of unavoidable sensitivity to the starting point, the optimization achievable being dependent on the specific sample configuration contributing to the constraint violation. This sensitivity is inherent to the problem as posed presently. Alternative formulations of the objective function can perhaps be visualized to reduce this sensitivity. It is of interest to note that the initiation point of run 1 represents the case corresponding to that of deploying only the linear classifier. In that case, we have 3-percent error in training sample set classification but no measure of an error bound for the operational phase. On the other hand, use of the NN classifier in a stand-alone mode requires 100 distance computations for each sample classification. But use of the composite classifier gives the opportunity to achieve 0-percent error outside the region of conflict (which corresponds to the case encountered most frequently) and a bounded (twice the Bayes error) error rate within the small region of conflict. In the latter (infrequent as it is) case, very few NN distances have to be computed as compared to the classical NN approach. This brings into sharp focus the advantages offered by the composite classifier system design.

VI. Discussion and Concluding Remarks

The main thrust of this study has been to put forth the concept of composite classifier systems as a means of achieving improved recognition system performance relative to that obtained with the classifier components (that make up this composite system) employed individually. This is illustrated by studying the case of the linear/NN classifier composite system. The feature space is partitioned to determine the domain of application of each component, and an optimality criterion is evolved to provide the rationale for the partitioning. This criterion ensures that the computational effort is less than that under NN classifier, and, at the same time, the recognition rate is equal to or better than under each of the components, thereby meeting the objective of improved recognition system performance. The optimality criterion is not global but specific to the classifier components chosen to formulate the composite system. However, these components are chosen on the basis of qualitative judgment about their individual performance capabilities, namely, low computational demands of the linear classifier and relatively unlimited scope for application (with bounded error rates) of the NN classifier. Nonparametric discriminants with prespecified nonlinear functional forms do not have such theoretically established error bounds but they do have higher computational demands compared to the linear classifier.

Although the resulting constrained optimization problem is open to solution by any suitable algorithmic tool, SWIFT was adapted in view of prior familiarity and successful experience with it elsewhere in other applications. This, however, does not rule out use of other alternatives. Thus this study opens up a new avenue in the field of classification system design by offering the concept of the composite classifier system as a potential tool for achieving improved recognition system performance.

References

[1] L. Kanal and B. Chandrasekaran, "On linguistic, statistical and mixed models for pattern recognition," in *Frontiers of Pattern Recognition*, S. Watanabe, Ed. New York: Academic Press, 1972.

[2] B. V. Dasarathy, "An integrated non-parametric sequential approach to multi-class pattern classification," *Int. J. Syst. Sci.*, vol. 4, pp. 449–457, May 1973.

[3] B. V. Dasarathy and B. V. Sheela, "Visiting nearest neighbors—A survey of nearest neighbor pattern classification techniques," in *Proc. Int. Conf. on Cybernetics and Society*, pp. 630–637, Sept. 1977.

[4] T. M. Cover and P. E. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inform. Theory*, vol. IT-13, pp. 21–27, Jan. 1967.

[5] R. L. Kashyap, "Algorithms for pattern classification," in *Adaptive Learning and Pattern Recognition Systems*, Fu and Mendel, Eds. New York: Academic Press, 1970, pp. 81–112.

[6] Y. C. Ho and R. L. Kashyap, "A class of iterative procedures for linear inequalities," *SIAM J. Contr.*, vol. 4, pp. 112–115, 1966.

[7] R. E. Warmack and R. C. Gonzalez, "An algorithm for the optimal solution of linear inequalities and its application to pattern recognition," *IEEE Trans. Comput.*, vol. C-22, pp. 1065–1075, Dec. 1973.

[8] G. Nagaraja and G. Krishna, "An algorithm for the solution of linear inequalities," *IEEE Trans. Comput.*, vol. C-23, pp. 421–427, Apr. 1974.

[9] B. V. Dasarathy, "DHARMA: Discriminant hyperplane abstracting residuals minimization algorithm for separating clusters with fuzzy boundaries," *Proc. IEEE*, vol. 64, pp. 823–824, May 1976.

[10] B. V. Sheela and P. Ramamoorthy, "SWIFT—A new constrained optimization technique," *Comput. Methods Appl. Mech. Eng.*, vol. 6, no. 9, pp. 309–318, 1975.

[11] B. V. Sheela, "Optimization of system reliability by sequential weight increasing factor technique," *IEEE Trans. Rel.*, vol. R-26, pp. 339–341, Dec. 1977.

[12] B. V. Sheela and B. V. Dasarathy, "OPAL: A new algorithm of optimal partitioning and learning in non-parametric unsupervised environments," *Int. J. Comput. Inform. Sci.*, vol. 8, no. 3, 1979.