Minimal Consistent Set (MCS) Identification for Optimal Nearest Neighbor Decision Systems Design

Belur V. Dasarathy

Abstract—**A new approach is presented in this study for tackling the problem of high computational demands of nearest neighbor (NN) based decision systems. The approach, based on the concept of an optimal subset selection from a given training data set, derives a consistent subset which is aimed to be minimal in size. This minimal consistent subset (MCS) selection, in contrast to most of the other previous attempts of this nature, leads to an unique solution irrespective of the initial order of presentation of the data. Further, consistency property is assured at every iteration. Also, unlike under most prior approaches, the samples are selected here in the order of significance of their contribution for enabling the consistency property. This provides insight into the relative significance of the samples in the training set. Experimental results based on a number of independent training and test data sets are presented and discussed to illustrate the methodology and bring to focus its benefits. These results show that the nearest neighbor decision system performance suffers little degradation when the given large training set is replaced by its much smaller MCS in the operational phase of testing with an independent test set. A direct experimental comparison with a prior approach is also furnished to further strengthen the case for the new methodology.**

I. PRIOR DEVELOPMENTS

Nearest Neighbor (NN) based decision systems, in the context of decision problems such as pattern classification, have been studied at length [1] over the past four decades. The main limitation on their usage in practice has been their computational demands in the operational phase. One of the concepts popularly advocated over the years to address this problem of high computational demands has been the selection of a representative subset of the training data so as to reduce the number of feature space distance computations in the operational phase. The other avenue explored in this context has been that of structuring the nearest neighbor search process so as to minimize the computational efforts. This is a purely computational strategy issue and involves no basic pattern recognition concepts. Both these have been discussed at length in the recent book on nearest neighbor techniques [1].

The first avenue, namely, selection of a design subset of prototypes from a given set of training data samples, has been addressed in the literature with varying degrees of success [1]–[15] from two different objectives. The first objective is of course the computational efficiency of the operational phase. The second one is that of editing the training sample set to make the resulting classification more reliable. The study reported here is mainly driven by the first objective while ensuring that the classification does not become less reliable during this process. Accordingly, this introductory review of earlier studies is limited to those with similar computational demand minimization objectives. The very first study of this kind was probably that of Hart [2] who in 1968, presented the "Condensed Nearest Neighbor Rule." In spite of the title, the study did not truly offer a new classification rule, but only a method of condensing or reducing the given training sample set into a smaller subset to be used with the classical NN rule. The study aims to preserve the integrity of the process by ensuring that the condensed set is consistent with the original set, i.e., all the original samples are correctly classified by the condensed set under the NN rule.

The method attempts to derive a minimal consistent subset without an exhaustive search. But, as admitted by the author, this goal of minimal subset is not realized. However, the exhaustive testing process ensures that the subset is indeed consistent. The method defines an initial condensed subset with a randomly picked representative prototype from each class. Using this subset, all the given samples are classified in an arbitrary order using the NN rule adding the misclassified samples onto the condensed subset. The process is repeated to ensure consistency and is continued till no more additions occur in a complete cycle. The method ensures consistency but not minimality of the condensed subset. In theory, it is possible here to end up with the original training set in its entirety. In practice, such a trivial result is unlikely and some savings in computations in the classification phase is indeed achievable. But, the method is very sensitive to the initial ordering of the input data.

In 1972, Swonger [3] presented the Iterative Condensation Algorithm (ICA). This approach, unlike the previous one [2], permitted both addition and deletion of samples to and from the condensed subset. The other advantages of the method are the accommodation of outliers or wrongly labeled samples, tolerance to identical valued samples with multiple labels, and convergence of the scheme, which permits early manual termination of the process prior to its automatic ending.

The "Reduced Nearest Neighbor Rule" of Gates [4], also presented in 1972, takes an opposite track to the CNN approach [2]. Instead of growing the condensed set from a null set [2], the reduced set is derived by iteratively contracting the given set. This is accomplished by retaining only those samples whose deletion would affect the correct classification of the remaining samples. This is again an iterative process since the processing of later samples affect earlier results. This iterative process has to provide for the possibility of reinsertion of dropped samples and has to be continued till stability is attained. Here, RNN set is always a subset of CNN set. Thus, if CNN does not contain the minimal consistent subset, then RNN also cannot contain such a minimal consistent subset. If however, CNN includes the minimal consistent subset, then RNN does result in the minimal consistent subset. The preprocessing costs for RNN implementation is far higher than that for the CNN. But the resultant RNN subset is always smaller than the CNN subset and hence in the operational phase, the RNN based classification system tends to be computationally more efficient than the CNN based one.

In 1974, Ullmann [5] presented two more methods which are essentially variations of the CNN rule [2]. The first method adds a dead zone to the minimum distance check during the process of identifying the condensed neighbor set. Whenever this dead zone is zero, this method effectively reduces to the CNN rule [2]. The second one, like the RNN approach [4], attempts to eliminate samples that are very far from the other classes, i.e., interior samples of the classes, by testing the samples in the order of increasing distances from all the other classes.

Also, in 1974, Chang [6] approached the problem of design data set reduction by viewing it as a problem of creating a customized design set rather than as one of just selecting a subset of the given set. The samples in his condensed set were therefore not directly identifiable members of the original set. Instead, the new pseudo-sample prototypes were generated by a process of merging nearest

Manuscript received December 2, 1991; revised April 6, 1993. A short preliminary version of this paper was presented at the 1991 IEEE/SMC International Conference on Systems, Man, and Cybernetics, held at University of Virginia, October 1991.

The author is with Dynetics, Inc., Huntsville, AL 35814-5050.

IEEE Log Number 9214589.

TABLE I
A DESCRIPTION OF THE TEST DATA SETS

Data Set No.	Number of Classes	Number of Features	Total No. of Samples	Class No.	Number of Samples
1	2	4	500	1	250
				2	250
2	2	4	500	1	250
				2	250
3	3	4	150	1	50
				2	50
				3	50
4	3	6	87	1	40
				2	34
				3	13
5	4	4	80	1	20
				2	20
				3	20
				4	20

TABLE II-A
RESULTS OF MCS SELECTION FOR DATA SET NO. 1 UNDER CHESSBOARD METRIC

Iteration Number	Number of Samples Selected		
	Class 1	Class 2	Total
1	7	4	11 (2.2 %)
2	6	4	10 (2.0 %)

TABLE II-B
RESULTS OF MCS SELECTION FOR DATA SET NO. 1 UNDER CITY-BLOCK METRIC

Iteration Number	Number of Samples Selected		
	Class 1	Class 2	Total
1	5	5	10 (2.0 %)
2	3	3	6 (1.2 %)

TABLE II-C
RESULTS OF MCS SELECTION FOR DATA SET NO. 1 UNDER EUCLIDEAN METRIC

Iteration Number	Number of Samples Selected		
	Class 1	Class 2	Total
1	6	4	10 (2.0 %)
2	5	4	9 (1.8 %)

neighbors of the same class as long as such merger did not increase the error rate. This approach, tends to decrease the size of the training set to a much greater extent than the other approaches considered hitherto. Ritter *et al.* [7] proposed in 1975 another method of selecting a subset of training samples aimed at satisfying the three criteria: i) consistency, ii) nearness to its own class sample than to any other, and iii) minimal of all solutions satisfying the previous two criteria. This method is more complex because of the need to satisfy these three criteria simultaneously.

Tomek [8], again in 1976, proposed two more modifications to the CNN approach [2]. The idea behind these is to explicitly aim at the retention of boundary samples by identifying the nearest neighbor of samples from the other classes. Ichino [9], in 1979, put forth a nonparametric multiclass pattern classifier involving generation of class regions. This is viewed as a training set reduction process and as such he compares his results with those obtained for CNN rule. In 1979, Gowda and Krishna [10] applied the concept of mutual neighborhood value (MNV), to the CNN approach. The MNV concept provides a measure of mutual nearness between NN pairs. In 1984, Fukunaga and Mantock [11] proposed a scheme for selecting a subset of representative samples from a given data set based on NN density estimates but made no reference to the earlier studies in this area.

Here, in this study, the emphasis is on the first perspective, namely, the minimization of computational loads. The main underlying concept of the new approach is that of the Nearest Unlike Neighbor Subset (NUNS) introduced recently [1] as a critical descriptor of training data sets for the NN based decision system. The details of the Minimal Consistent Subset (MCS) methodology and its development from the NUNS concept are presented in Section II followed by the algorithmic procedure in Section III. A description of the test data sets is given in Section IV. This is followed by the presentation and discussion of test results in Section V. The scope for extension to the domain of k-NN rules is discussed briefly in Section VI. The last section presents some concluding comments.

II. MCS METHODOLOGY

As stated in the previous section, MCS selection is based on the concept of NUNS, the Nearest Unlike Neighbor Subset [1], which can be looked upon as an optimal descriptor of the inter-class boundaries. The NUN subset is defined as the unique set of all samples which are the nearest unlike neighbors of one or more of the given samples. The properties of NUN set and other related topics are covered in [1]. Based on this concept, we can see that for every given sample, the sufficient condition for its correct classification, i.e., for consistency, is the presence within MCS a sample from its own class that is closer than its NUN (nearest unlike neighbor). Obviously, many samples independently satisfy this sufficiency condition for each given sample under consideration. This can be looked upon as a vote of confidence cast by the given sample and received by such closer-than-NUN samples. The sample with the most such votes, i.e., the sample that satisfies the consistency conditions for most number of samples, therefore represents the prime candidate for inclusion in MCS. Once this is picked, all the samples which were the voters contributing to the selection of the candidate for MCS can be disregarded from further consideration (a typical post-election phenomenon) and the vote counts of other candidates are reduced to reflect this. The candidate with the maximum votes after this update becomes the next most effective MCS sample. This process is repeated till all the voters have been taken into account, i.e., till full consistency is achieved. It is of course possible that in some cases the samples may have only one vote, i.e., of itself. In such cases, these automatically become MCS candidates. It is also possible that the voters to another sample may themselves become candidates for MCS.

Once a candidate MCS set (based on the NUN distances computed over the entire set) has been identified, it is necessary to reexamine the problem as the effective NUN distances are now likely to be larger than before as some NUNs are no longer in the subset under consideration. Thus there is now scope for reducing the candidate MCS further. However, for the process to be monotonically reducing (i.e., we should not bring in new samples which were not required in the previous iteration) we have to ensure that the candidate list will only include samples (other than the last MCS candidates) that will not create any new inconsistencies (see step 5 under the algorithmic procedure). This process is thus repeated until the set size can no longer be reduced. The method, unlike most previous methods, is insensitive to the initial order of the data since the initial selection of the samples to the candidate set is not random. Instead, there is a sound conceptual basis for the selection process. This also assures consistency at every iteration. This in essence represents the core of the advancements made by this method over earlier attempts. Also, since the candidates to MCS are selected on the basis of the extent of their contribution to the overall consistency property, it provides insight into the relative significance of the samples within the input

TABLE III-A
RESULTS OF MCS SELECTION FOR DATA SET NO. 2 UNDER CHESSBOARD METRIC

Iteration Number	Number of Samples Selected		
	Class 1	Class 2	Total
1	58	57	115 (23.0 %)
2	49	54	103 (20.6 %)
3	48	52	100 (20.0 %)

TABLE III-B
RESULTS OF MCS SELECTION FOR DATA SET NO. 2 UNDER CITY-BLOCK METRIC

Iteration Number	Number of Samples Selected		
	Class 1	Class 2	Total
1	53	57	110 (22.0 %)
2	50	54	104 (20.8 %)
3	49	54	103 (20.6 %)
4	49	53	102 (20.4 %)

TABLE III-C
RESULTS OF MCS SELECTION FOR DATA SET NO. 2 UNDER EUCLIDEAN METRIC

Iteration Number	Number of Samples Selected		
	Class 1	Class 2	Total
1	57	55	112 (22.4 %)
2	49	50	99 (19.8 %)
3	49	49	98 (19.6 %)

training set. The methodology as outlined here is currently applicable to the single k-NN rule scenario. However, conceptually this can be extended to other scenarios, such as k-NN rules, as well, as discussed in a later section.

III. ALGORITHMIC PROCEDURE

Step 1 Define an initial consistent set to be the given training data set, since the given set is by definition consistent with itself.

Step 2 For a specific sample in the given training data set, determine the nearest sample distance among all the samples from all classes other than its own in the consistent set, i.e., identify and store the Nearest Unlike Neighbor (NUN) distance of the sample from the consistent set.

Step 3 For this same sample, Identify all the neighboring samples from its own class in the given data set which are closer than this NUN distance and cast an approval vote to each of these samples in the given set by incrementing the corresponding vote registers, while noting this voter's (sample) identity by updating the corresponding voter lists.

Step 4 Repeat Step 2 and 3 for all samples in the given training set, which results in a list of the number of votes received by each sample in the given set along with the records of identity of its voters.

Step 5 Create a potential candidate consistent set consisting of all samples in the given set which are either (a) already present in the current consistent set or (b) whose inclusion will not create an inconsistency; i.e., the sample should not be nearer to any member of any other class than that member's current NUN distance. In the first iteration, the entire consistent set (i.e., the given set) remains as the candidate consistent set as all samples satisfy condition (a).

Step 6 Identify the most voted sample in this candidate consistent list and designate it as a member of a newly selected consistent set and identify all of its contributing voters.

TABLE IV-A
RESULTS OF MCS SELECTION FOR DATA SET NO. 3 UNDER CHESSBOARD METRIC

Iteration Number	Number of Samples Selected			
	Class 1	Class 2	Class 3	Total
1	1	10	9	20 (13.3 %)
1	1	9	9	19 (12.7 %)

TABLE IV-B
RESULTS OF MCS SELECTION FOR DATA SET NO. 3 UNDER CITY-BLOCK METRIC

Iteration Number	Number of Samples Selected			
	Class 1	Class 2	Class 3	Total
1	1	8	9	18 (12.0 %)
2	1	8	8	17 (11.3 %)

TABLE IV-C
RESULTS OF MCS SELECTION FOR DATA SET NO. 3 UNDER EUCLIDEAN METRIC

Iteration Number	Number of Samples Selected			
	Class 1	Class 2	Class 3	Total
1	1	6	9	16 (10.7 %)
2	1	6	8	15 (10.0 %)

Step 7 Delete these voters from all the voter lists wherein they currently appear and correspondingly decrement the appropriate vote counts.

Step 8 Repeat Step 6 and Step 7 till all the voters have been accounted for by the selected consistent set.

Step 9 Now with this selected consistent set, the NUN distances of the input samples are likely to be greater than before as some of the original NUN samples may no longer be in the selected consistent set. Accordingly, repeat Step 2 using this selected consistent set to determine the NUN distance thresholds for each sample in the given set.

Step 10 Repeat Step 3 through 8 using all the samples in the given set to identify a new consistent set. This process of recursive application of step 2 through 8 is continued till the selected set is no longer getting smaller. It is easy to see that under this procedure this final subset remains consistent, i.e., is able to classify all samples in the original set correctly.

IV. TEST DATA SET DESCRIPTION

As shown in Table I, five independent data sets were used to test the new methodology. The selection was motivated by the desirability to have broad spectrum of variations in all the different parameters of relevance such as number of classes, number of features, and number of samples. The first is a two-class, four-dimensional data set consisting of 250 samples from each of the two well-separated classes. The second set is also of similar size and dimensionality but with more overlap between the classes. These two data sets were generated in-house under some discrimination simulation studies. The third one is the now famous Iris data set [12] consisting of three classes each with 50 four-dimensional samples. The fourth one is a three class, six-dimensional Fossil data set [12] with 87 samples in total. The fifth one is a real world textural feature data set [13] extracted from cell imagery data. The spectrum of data sets used in the experiments help in assessing the possible effects of data dimensionality, multi-class effects and such other factors.

V. EXPERIMENTAL RESULTS

The results corresponding to these five data sets are shown in the set of Tables II-A, B, and C through VI-A, B, and C. Three alternative

TABLE V-A
RESULTS OF MCS SELECTION FOR DATA SET NO. 4 UNDER CHESSBOARD METRIC

Iteration Number	Number of Samples Selected			
	Class 1	Class 2	Class 3	Total
1	6	1	6	13 (14.9%)
2	5	1	5	11(12.6%)

TABLE V-B
RESULTS OF MCS SELECTION FOR DATA SET NO. 4 UNDER CITY-BLOCK METRIC

Iteration Number	Number of Samples Selected			
	Class 1	Class 2	Class 3	Total
1	2	2	2	6 (6.9%)
2	2	1	1	4 (4.6%)

TABLE V-C
RESULTS OF MCS SELECTION FOR DATA SET NO. 4 UNDER EUCLIDEAN METRIC

Iteration Number	Number of Samples Selected			
	Class 1	Class 2	Class 3	Total
1	3	2	4	9(10.3%)
2	2	1	2	5(5.7%)

distance metrics, the chessboard [14], city-block, and Euclidean, were employed to evaluate the sensitivity of the approach to the type metric used. Table parts -A, -B, and -C, respectively, present the results corresponding to these three metrics.

In the first data set (Tables II-A, B, and C), the application of the MCS selection technique resulted in minimal consistent subsets of 10, 6, and 9 samples respectively under the three metrics. This corresponds to no more than 2% of the input set and thus represents an impressive computational demand optimization. In the second case (Tables III-A, B, and C), the resulting subset was much larger and consisted of 100, 102, and 98 samples under the three metrics respectively, i.e., about 20% of the original set. For the Iris data set (Tables IV-A, B, and C), the results were in between with 19, 17, and 15 samples being chosen for MCS under the three metrics, i.e., about 10–12% of the original data set. For the Fossil data set (Tables V-A, B, and C), the resulting MCS size was 11, 4, and 5 respectively for the three metrics. This is once again an example of a very significant computational demand minimization. For the image textural data set , the resulting MCS size was of the order of 8 to 12%, showing that even with the increase in the number of classes, the reduction in sample set size was still significant.

These results indicate that the method is not very sensitive to the specific metric chosen, although in some cases the chessboard metric tends to result in less minimization. This is mainly because the chessboard metric effectively represents a single feature at a time and hence the class overlap can be considerably higher under this metric. Under all the three metrics, most of the reduction is achieved at the very first iteration and since consistency is guaranteed at each iteration, one could halt the process after the first iteration for all practical purposes. Also, the reduction in set size is more or less equal across the classes for each of these data sets, except for class 1 in the case of the Iris data set. This is because class 1 of Iris data set is significantly farther from the other two classes, which are closer to each other and hence require more number of representative samples to achieve consistency. From the view point of the computational demands, it is interesting to determine how much loss in the consistency property results from an incomplete set. Fig. 1 shows the variation in the extent of consistency achieved by the selected subset as a function of the size of the selected subset for the data set 1 under the Euclidean norm. As is to be expected from

TABLE VI-A
RESULTS OF MCS SELECTION FOR DATA SET NO. 5 UNDER CHESSBOARD METRIC

Iteration Number	Number of Samples Selected				
	Class 1	Class 2	Class 3	Class 4	Total
1	4	4	3	1	12 (15.0%)
2	3	3	2	1	9 (11.3%)

TABLE VI-B
RESULTS OF MCS SELECTION FOR DATA SET NO. 5 UNDER CITY-BLOCK METRIC

Iteration Number	Number of Samples Selected				
	Class 1	Class 2	Class 3	Class 4	Total
1	3	3	1	1	8(10.0%)
2	3	2	1	1	7 (8.8%)

TABLE VI-C
RESULTS OF MCS SELECTION FOR DATA SET NO. 5 UNDER EUCLIDEAN METRIC

Iteration Number	Number of Samples Selected				
	Class 1	Class 2	Class 3	Class 4	Total
1	3	3	1	1	8(10.0%)
2	2	3	1	1	7 (8.8 %)
3	2	2	1	1	6 (7.5%)
4	2	2	1	1	6 (7.5 %)
5	2	2	1	1	6 (7.5 %)

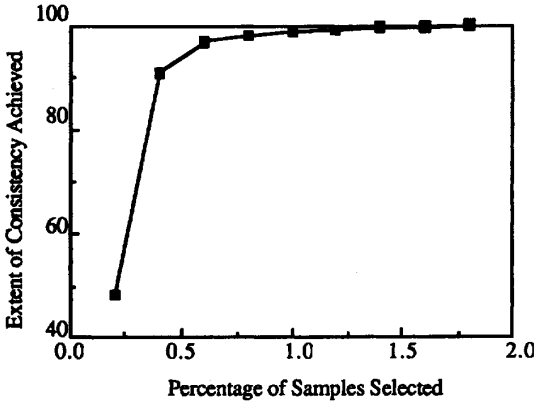

Fig. 1 Consistency vs. percentage of samples selected for data set no. 1.

the logic underlying the selection process, the contribution of the selected samples to the overall consistency property monotonically decreases with the initial sample offering the most. As such the total consistency offered by the subset increases faster initially but slows down gradually until hundred percent consistency is achieved. This is further confirmed by the results shown in Figs. 2 through 5 for the other four data sets under the Euclidean norm. Except for data set no. 2, close to 100% consistency is achieved in all cases with a very small percentage of samples. Results are more or less similar under the chessboard and city-block distance metrics also.

While consistency is a measure of performance relative to the training set, a true test of the subset selection process would be to determine its effect on the classification of a separate test data set. This can be done by comparing the classification performances obtained on an independent test data set using the original training set and the selected subset. Such a test was carried out using the same five data sets by dividing them into two equal halves, with one half serving as training set and the other as test set and later switching

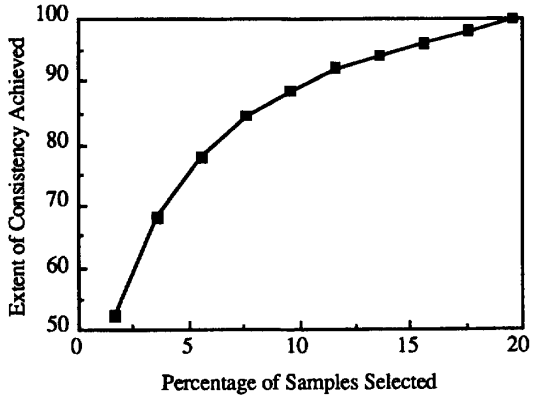

Fig. 2. Consistency vs. percentage of samples selected for data set no. 2.

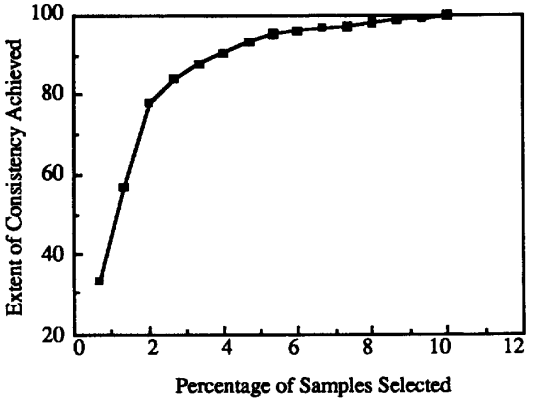

Fig. 3. Consistency vs. percentage of samples selected for data set no. 3.

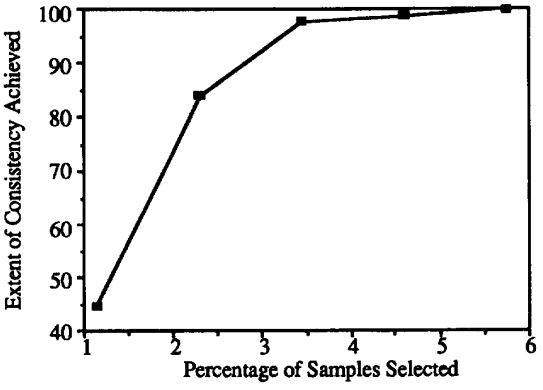

Fig. 4. Consistency vs. percentage of samples selected for data set no. 4.

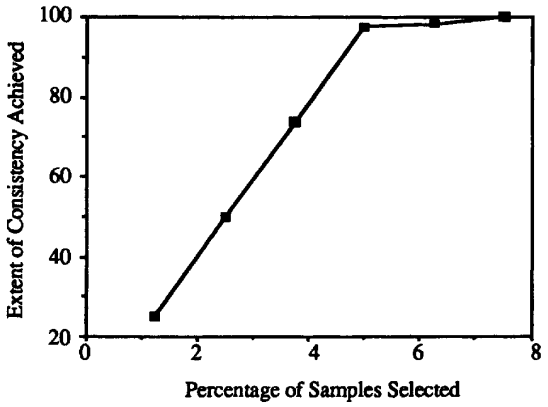

Fig. 5. Consistency vs. percentage of samples selected for data set no. 5.

TABLE VII-A
FULL SET AND MCS BASED CLASSIFICATION RESULTS USING DATA SET NO. 1

Test Set (Training Set)	Full Set Size	Full Set Efficiency	MCS Size	MCS Efficiency
2 (1)	250	100.0	4	97.20
1 (2)	250	100.0	5	100.00

TABLE VII-B
FULL SET AND MCS BASED CLASSIFICATION RESULTS USING DATA SET NO. 2

Test Set (Training Set)	Full Set Size	Full Set Efficiency	MCS Size	MCS Efficiency
2 (1)	250	88.80	49	88.80
1 (2)	250	86.80	5	84.00

TABLE VII-C
FULL SET AND MCS BASED CLASSIFICATION RESULTS USING DATA SET NO. 3

Test Set (Training Set)	Full Set Size	Full Set Efficiency	MCS Size	MCS Efficiency
2 (1)	150	94.67	9	92.00
1 (2)	150	94.67	11	92.00

TABLE VII-D
FULL SET AND MCS BASED CLASSIFICATION RESULTS USING DATA SET NO. 4

Test Set (Training Set)	Full Set Size	Full Set Efficiency	MCS Size	MCS Efficiency
2 (1)	43	95.00	3	93.33
1 (2)	44	94.44	6	100.0

TABLE VII-E
FULL SET AND MCS BASED CLASSIFICATION RESULTS USING DATA SET NO. 5

Test Set (Training Set)	Full Set Size	Full Set Efficiency	MCS Size	MCS Efficiency
2 (1)	40	97.50	4	97.50
1 (2)	40	100.0	6	97.50

their roles. The results of classification of the test set corresponding to the cases of using the entire training set as well as its MCS are tabulated in Table VII-A through E. For three of the data sets (nos. 1, 2, and 5), there is no loss whatsoever in classification performance (attributable to the reduction in the size of the training set) under one of the training/test set dichotomy. Under the other dichotomy, there is only a slight marginal reduction. For data set no. 3, there is an identical marginal decrease under both dichotomies. For data set no. 4 in one case there is again a marginal loss and in the other there is in fact a gain of over 5%. This increase is of course not necessarily significant in a statistical sense except perhaps to confirm the fact that the observed loss in the other cases may also not be necessarily significant in the statistical sense. This shows that the huge savings

TABLE VIII
A COMPARISON OF THE COMPUTATIONAL LOADS UNDER CNN RELATIVE TO MCS

Data Set No.	CNN Set Size	MCS Size	Excess Computational Demands under CNN (relative to MCS)
1	9	9	0.00 %
2	124	98	26.53 %
3	24	15	60.00 %
4	10	5	100.00 %
5	11	6	83.33 %

in the computational effort (ranging from about 98% for set no. 1 to about 80% for set no. 2 with all the others somewhere in between) is well worth the possible small loss in the recognition accuracy.

As a final measure of evaluation of the new methodology, experimental comparison of MCS with the CNN Rule was carried out (using Euclidean norm) and the results thereof are tabulated in Table VIII. As shown therein, except for data set no. 1 (which is a essentially a completely separated data set and hence requires very few samples to describe the separation surface), in every other case the excess computational load of using CNN over MCS is very significant. This convincingly demonstrates the benefits to be accrued by using the MCS methodology. Likewise, comparisons with other methods can also be conceived, but were beyond the resources available for this study.

VI. EXTENSION TO THE k-NN RULES DOMAIN

Although the single NN rule was employed in these experiments, the method can conceivably be extended to the k-NN rule domain. Under this scenario, instead of requiring only one sample (from its own class) nearer than its NUN to be present in MCS for each input sample, we would, for sufficiency, need at least $(k/2+1)$ number of samples within the NUN distance to be present in MCS for each input sample. The steps 1 through 5 listed under the algorithmic procedure (Section III) would remain essentially the same. The steps 6 and 7 will have to be suitably modified so that the optimal combination that delivers the maximum consistency is identified at each stage. For example, for the case $k = 3$, we will have to identify the pair of samples that together satisfy the consistency property for the largest number of input samples and repeat the process to identify each addition to the candidate set. The remaining steps once again would more or less be similar to that under the current procedure as described for the single-NN rule. One could also visualize extension to other scenarios such as the weighted k-NN rule domain [1].

All of these potential extensions, although conceptually feasible, make the combinatorial search procedures of identifying the MCS much more complex. It is likely that the resultant consistent sets under these complex rules would be larger than under the single-NN rule, thus reducing the gains in terms of computational demand optimization. But this has to be weighed against the possible benefits of robustness that could result by using such rules. The possible extensions of the method therefore pose trade-off problems that need to be addressed for evaluating the practicality of this methodology under such complex NN-rule environments. These extensions are accordingly considered outside the scope of the present study and will be reported in due course at other appropriate forums.

VII. CONCLUDING COMMENTS

The MCS method offers an effective tool for minimizing the computational demands of NN based decision systems. This is accomplished by development of a rational procedure for the selection of an optimal subset out of the available training sample set. The main features of this approach as compared to prior approaches are as follows:

- The results are independent of the order of presentation of the data and hence the selected subset is always unique both in terms of the number and identity of the selected samples.
- Consistency property is always assured at every stage of the iterative process, which permits the iterative process to be manually terminated when desired without loss of the consistency property.
- Since almost all of the minimization that can be expected for any specific data set is achieved in the first couple of iterations, there is no compelling need to go beyond the first one or two iterations.
- Also, with very little sacrifice in the consistency criterion, one can achieve further savings in the computations by picking a subset of desired size (starting with the first) of the MCS. This is possible, since samples in MCS are selected in the order of their contribution to the consistency property rather than randomly as is the case under most other approaches.
- A truly encouraging aspect of these results is the negligible loss in recognition efficiency when the full training set is replaced by its MCS in the operational phase of testing an independent test data set.
- A direct comparison with the popular CNN approach further confirms the efficacy of the new approach.

Thus, it is clear that the new approach offers significant advancement in the state of the art in the area of computational demand optimization as applied to NN-based decision systems. While no formal mathematical proof of convergence of the iterative process has been established by this study, the iterative algorithmic procedure ensures that such is indeed the case and all of the experiments have also further confirmed this fact. With regard to the issue of attaining true minimality of the size of subset derived under this approach, although experimental evidence strongly suggests this to be the case, a formal mathematical analysis would be required to substantiate this claim. However, this was deemed beyond the scope of the support available for this study. The potential for expanding the MCS concept to the domain of learning and recognition in imperfectly supervised and/or partially exposed environments [1] as well as under fuzzy teacher environments [15] is currently under exploration and will be reported in due course.

ACKNOWLEDGMENT

The author gratefully acknowledges the support and encouragement provided to this research effort by Mr. Tom Baumbach, Executive Vice-President, Dynetics, Inc., Huntsville, AL.

REFERENCES

[1] B. V. Dasarathy, *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques*. Los Alamitos, CA: IEEE Computer Society Press, 1991.
[2] P. E. Hart, "The condensed nearest neighbor rule," *IEEE Trans. Information Theory*, vol. IT-14, no. 3, pp. 515–516, May 1968.
[3] C. W. Swonger, "Sample set condensation for a condensed nearest neighbor decision rule for pattern recognition," *Frontiers of Pattern Recognition*, S. Watanabe, ed. New York: Academic Press, pp. 511–519, 1972.
[4] G. W. Gates, "The reduced nearest neighbor rule," *IEEE Trans. Information Theory*, vol. IT-18, no. 3, pp. 431–433, May 1972.
[5] J. R. Ullmann, "Automatic selection of reference data for use in a nearest neighbor method of pattern classification," *IEEE Trans. Information Theory*, vol. IT-20, no. 4, pp. 541–543, July 1974.
[6] C. L. Chang, "Finding prototypes for nearest neighbor classifiers," *IEEE Trans. Computers*, vol. C-23, no. 11, pp. 1179–1184, Nov. 1974.

[7] G. L. Ritter, H. B. Woodruff, S. R. Lowry, and T. L. Isenhour, "An algorithm for a selective nearest neighbor decision Rule," *IEEE Trans. Information Theory*, vol. IT-21, no. 6, pp. 665–669, Nov. 1975.
[8] I. Tomek, "Two modifications of CNN," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-6, no. 11, pp. 769–772, Nov. 1976.
[9] M. Ichino, "A nonparametric multiclass pattern classifier," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-9, no. 6, pp. 345–352, June 1979.
[10] K. C. Gowda and G. Krishna, "The condensed nearest neighbor rule using the concept of mutual nearest neighborhood," *IEEE Trans. Information Theory*, vol. IT-25, no. 4, pp. 488–490, July 1979.
[11] K. Fukunaga and J. M. Mantock, "Nonparametric data reduction," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. PAMI-6, no. 1, pp. 115–118, Jan. 1984.
[12] Y. T. Chien, *Interactive Pattern Recognition*. New York: Marcel Dekker, Inc., pp. 223–230, 1978.
[13] B. V. Dasarathy and E. B. Holder, "Image characterizations based on joint gray level-run length distributions," *Pattern Recog. Lett.*, vol. 12, no. 8, pp. 497–502, Aug. 1991.
[14] F. Rhodes, "Some characterizations of the chessboard metric and the city block metric," *Pattern Recog. Lett.*, vol. 11, no. 10, pp. 669–675, Oct. 1990.
[15] B. V. Dasarathy, "Fuzzy learning under and about an unfamiliar fuzzy teacher," in *NAFIPS'92, Proc. North American Fuzzy Information Processing Society Conf.*, pp. 368–377, Dec. 1992.

Collision Avoidance of Two General Robot Manipulators by Minimum Delay Time

Cheol Chang, Myung Jin Chung, and Bum Hee Lee

***Abstract*—A simple time delay method for avoiding collisions between two general robot arms is proposed. Links of the robots are approximated by polyhedra and the danger of collision between two robots is expressed by distance functions defined between the robots. The collision map scheme, which can describe collisions between two robots effectively, is adopted. The minimum delay time value needed for collision avoidance is obtained by a simple procedure of following the boundary contour of collision region on collision map. To demonstrate the effectiveness of the proposed time delay method, a computer simulation study is shown where a collision is likely to occur realistically.**

I. Introduction

Industrial robots have made a significant contribution toward automating the manufacturing processes. The efficient use of robots shows productivity increase, production cost reduction, and product quality improvement. However, most robots currently in use perform simple repetitive jobs, such as pick-and-place, machine loading and unloading, spray painting, and spot welding.

Only one robot in a common work space limits the classes of tasks that can be performed. Two or more robots in a work space can improve potential application area of robots. Multiple robots can be used to accomplish a task where each performs its own subtask in parallel, and to save the production time. Also, multiple robots can accomplish complex tasks that can not be performed by a single robot such as transporting an object beyond the payload capability of a single robot. Among the previous applications of multiple robots, the parallel tasking feature is only an example especially in industry aiming to mass production. However, at present, the parallel tasking is not fully utilized since there is no practical methodology that can make several robots operate safely in a common workspace. In the case that more than one robot operate simultaneously in a common workspace, the problem of avoiding potential collisions between the robots should be considered very carefully.

To solve the collision avoidance problem, zone-blocking methods have been proposed. In these methods only one robot operates at a time. So, this semaphore mechanism is not efficient because of not providing the parallel tasking feature. Besides zone-blocking methods some collision avoidance methods [1]–[5] have been proposed for multiple robots. These methods can be divided into two categories: 1) time adjusting methods while maintaining the given geometric path and 2) trajectory modification methods which modifies given geometric path. The former adjusts the time evolution representing the moving speed of robots while the geometrical paths of the robots are fixed. The robot path, which guarantees a robot not to collide with stationary obstacles, can be obtained using some existing methods [6]–[10]. One of the major features of time adjusting approaches is that the number of variables to be considered for collision avoidance does not exceed the number of robots because one variable, usually the time, is enough to express the moving speed for each robot. For instance, in the case of two robots, at most two variables are needed for solving the collision avoidance problem. This fact suggests that a collision avoidance problem in multiple robots can be easily solved comparing with a collision avoidance problem for a single robot and stationary obstacles which requires at least three variables in a three dimensional work space.

Lee *et al.* [2] presented several time adjusting methods for two robots using a collision map. In their paper, only a wrist of each robot is treated as a possible collision obstacle and modeled by a sphere. A collision map is used to describe potential collisions between two robots efficiently under the condition that given geometrical paths for two robots are fixed. However, the collisions in three-dimensional work space are transformed into collision region(s) in the map and this transformation usually requires an excessive computational effort. Furthermore, the collision region(s) in the map is approximated by box(es) to determine the departure time of one robot. Therefore, this approximation of collision region(s) results in unnecessary extra time delay.

This paper proposes an effective collision avoidance method for two general robot manipulators which are approximated by polyhedra as an extension of Lee *et al.* The proposed method determines the minimum time delay needed for avoiding collisions between two general robot manipulators using distance functions.

Basically the computational scheme for obtaining the delay time adopts the concept of the collision map which represents the region corresponding to collisions between two robots. To obtain the collision-free minimum delay time, a scheme which follows the boundary contour of the collision region in a collision map is proposed. This scheme only checks a part of the boundary contour of the collision region and the TLVST (traveling length versus sampling time) curve conceptually. Due to the computational simplicity, the overall procedure is very simple. Also, this method of avoiding collisions through time delay is relatively easy to implement, because the geometrical paths and time evolution of two robot manipulators

Manuscript received January 9, 1992; revised April 6, 1993.
C. Chang is with the Samsung Advanced Institute of Technology, Suwon, Kyung Ki-do, Korea.
M. J. Chung is with the Department of Electrical Engineering, Korea Advanced Institute of Science and Technology, Taejon 305–701, Korea.
B. H. Lee is with the Department of Control and Instrumentation Engineering, Seoul National University, Seoul, Korea.
IEEE Log Number 9214588.