of torque minimization, a smaller value of μ is preferable, and as a tradeoff, a larger critical area has to be prescribed [e.g., $\eta = 0.8$ when selecting $\mu = 2.0$ as in Fig. 9(c)]. In summary, performance of the presented examples demonstrates the effectiveness of the proposed dual neural network model (12) on the path-following and torque minimization task of redundant manipulators with limited joint ranges.

VI. Concluding Remarks

The one-layer dual neural network model provides a new parallel distributed computational approach to real-time torque minimization of limited-joint-range redundant manipulators in real-time. Compared with other studies on torque optimization, the proposed formulation resolves redundancy at the acceleration level and simultaneously considers the joint limit avoidance. Different from other recurrent neural network approaches, the proposed dual neural network is developed for avoiding robot joint limits during the path-following torque optimization task. Future works include the neural network implementation on a dedicated hardware such as ASIC, and then experimental verifications on different manipulators.

References

[1] L. Sciavicco and B. Siciliano, *Modeling and Control of Robot Manipulators*. New York: Springer-Verlag, 2000.

[2] Z. Mao and T. C. Hsia, "Obstacle avoidance inverse kinematics solution of redundant robots by neural networks," *Robotica*, vol. 15, pp. 3–10, 1997.

[3] T. Yoshikawa, "Manipulability of robot mechanisms," *Int. J. Robot. Res.*, vol. 4, no. 2, pp. 3–9, 1985.

[4] A. S. Deo and I. D. Walker, "Minimum effort inverse kinematics for redundant manipulators," *IEEE Trans. Robot. Automat.*, vol. 13, pp. 767–775, Oct. 1997.

[5] J. M. Hollerbach and K. C. Suh, "Redundancy resolution of manipulators through torque optimization," *IEEE Trans. Robot. Automat.*, vol. RA-3, pp. 308–315, 1987.

[6] A. Nedungadi and K. Kazerounian, "A local solution with global characteristics for joint torque optimization of a redundant manipulator," *J. Robot. Syst.*, vol. 6, no. 5, pp. 631–654, 1989.

[7] H. J. Kang and R. A. Freeman, "Null space damping method for local joint torque optimization of redundant manipulators," *J. Robot. Syst.*, vol. 10, pp. 249–270, 1993.

[8] S. Ma, "A balancing technique to stabilize local torque optimization solution of redundant manipulators," *J. Robot. Syst.*, vol. 13, no. 3, pp. 177–185, 1996.

[9] ——, "A new formulation technique for local torque optimization of redundant manipulators," *IEEE Trans. Ind. Electron.*, vol. 43, pp. 462–468, 1996.

[10] W. S. Tang and J. Wang, "Two recurrent neural networks for local joint torque optimization of kinematically redundant manipulators," *IEEE Trans. Syst., Man, Cybern. B*, vol. 30, pp. 120–128, Jan. 2000.

[11] K. Kazerounian and A. Nedungadi, "Redundancy resolution of robotic manipulators at the acceleration level," in *Proc. 7th World Congress Theory of Machines and Mechanisms (IFToMM)*, Sevilla, Spain, 1987.

[12] K. C. Suh and J. M. Hollerbach, "Local versus global torque optimization of redundant manipulators," in *Proc. IEEE Int. Conf. Robotics and Automation*, Raleigh, NC, Mar. 1987.

[13] J. Wang, Q. Hu, and D. Jiang, "A Lagrangian network for kinematic control of redundant manipulators," *IEEE Trans. Neural Networks*, vol. 10, pp. 1123–1132, Sept. 1999.

[14] H. Ding and S. K. Tso, "A fully neural-network-based planning scheme for torque minimization of redundant manipulators," *IEEE Trans. Ind. Electron.*, vol. 46, pp. 199–206, 1999.

[15] W. S. Tang and J. Wang, "A recurrent neural network for minimum infinity-norm kinematic control of redundant manipulators with an improved problem formulation and reduced architectural complexity," *IEEE Trans. Syst., Man, Cybern. B*, vol. 31, pp. 98–105, Jan. 2001.

[16] Y. Xia and J. Wang, "A dual neural network for kinematic control of redundant robot manipulators," *IEEE Trans. Syst., Man, Cybern. B*, vol. 31, pp. 147–154, Jan. 2001.

[17] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming—Theory and Algorithms*. New York: Wiley, 1993.

[18] D. P. Bertsekas, *Parallel and Distributed Computation: Numerical Methods*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[19] O. L. Mangasarian, "Solution of symmetric linear complementarity problems by iterative methods," *J. Optim. Theory Applicat.*, vol. 22, no. 2, pp. 465–485, 1979.

[20] W. Li and J. Swetits, "A new algorithm for solving strictly convex quadratic programs," *SIAM J. Optim.*, vol. 7, no. 3, pp. 595–619, 1997.

[21] J.-S. Pang and J.-C. Yao, "On a generalization of a normal map and equation," *SIAM J. Control Optim.*, vol. 33, pp. 168–184, 1995.

[22] Y. Xia and J. Wang, "Global exponential stability of recurrent neural networks for solving optimization and related problems," *IEEE Trans. Neural Networks*, vol. 11, pp. 1017–1022, 2000.

[23] ——, "Global asymptotic and exponential stability of a dynamic neural system with asymmetric connection weights," *IEEE Trans. Automat. Contr.*, vol. 46, pp. 635–638, 2001.

[24] P. I. Corke and B. Armstrong-Helouvry, "A search for consensus among model parameters reported for the PUMA 560 robot," in *Proc. IEEE Int. Conf. Robotics and Automation*, vol. 2, 1994, pp. 1608–1613.

A Merge-Based Condensing Strategy for Multiple Prototype Classifiers

Ramón A. Mollineda, Francesc J. Ferri, and Enrique Vidal

***Abstract*—A class-conditional hierarchical clustering framework has been used to generalize and improve previously proposed condensing schemes to obtain multiple prototype classifiers. The proposed method conveniently uses geometric properties and clusters to efficiently obtain reduced sets of prototypes that accurately represent the data while significantly keeping its discriminating power. The benefits of the proposed approach are empirically assessed with regard to other previously proposed algorithms which are similar in their foundations. Other well-known multiple prototype classifiers have also been taken into account in the comparison.**

***Index Terms*—Clustering, condensing, multiple prototypes, nearest neighbors (NNs).**

I. Introduction

One of the best known and most extensively studied family of pattern classifiers is the k-nearest neighbor (NN) rule. Given a training set of previously labeled samples (or prototypes) and an unknown sample x, the k-NN rule assigns the most frequently represented class-label among the k closest prototypes to x. In spite of its conceptual simplicity, the rule is asymptotically optimal in the Bayes sense [1], provided there is an arbitrarily large number of prototypes n available so that k can be taken as large as necessary, while keeping the ratio k/n arbitrarily small which, in turn, implies that the neighbors are infinitely close to x. A trivial consequence of the large size of the sets of prototypes is the computational burden this searching problem implies. Another very important drawback comes from the erroneously labeled

Manuscript received March 21, 2002. This work was supported in part by Spanish project TIC2000-1703-C03-03. This paper was recommended by Associate Editor W. Pedrycz

R. A. Mollineda and E. Vidal are with the Institut Tecnològic d'Informàtica, Universitat Politècnica de València, 46071 València, Spain.

F. J. Ferri is with the Departamento d'Informàtica, Universitat de València, 46100 Burjassot, Spain.

Publisher Item Identifier S 1083-4419(02)06293-3.

or noisy prototypes, which may lead to arbitrarily large deviations from the asymptotically optimal results which could be expected.

Several different ways of circumventing these drawbacks have been proposed. In this context, *prototype selection* tries to modify the initially given set of prototypes, in order to reduce its size as well as improve classification performance. This family of methods can be divided into *editing* and *condensing* techniques [2]. The editing technique aims at removing outliers and prototypes which are placed at the overlap among classes, producing well-clustered groups of homogeneous prototypes that lead to optimal (in the Bayes sense) 1-NN classification. On the other hand, condensing algorithms try to find a significantly reduced set of prototypes the 1-NN results of which are as close as possible to those obtained using all of the original prototypes. Because of its goal, pure condensing algorithms cannot properly generalize when the original 1-NN classifier is far from optimal (mainly because of overlapping). Nevertheless, generalization can be achieved in several different ways, e.g., by previously using editing [2].

According to the way in which prototypes are obtained, these algorithms are separated into *selection* techniques [3]–[5], in which the resulting prototypes are taken from the original set, and *replacement* techniques [6]–[8], in which resulting prototypes are built and may be different from any prototype in the original set.

A novel approach for obtaining a condensed set of prototypes by replacement from an initial set is presented in this paper. A hierarchical agglomerative framework is used along with a convenient criterion to keep discriminating power while obtaining representative (and relatively reduced) sets of prototypes [9]. This scheme constitutes a generalization of previously presented approaches [6], [8].

II. Prototype Replacement Algorithms

Prototype replacement algorithms are usually considered as objectively superior to selection algorithms because there is no restriction upon the position of the prototypes finally obtained. Although this is only true for vectorial representation spaces, the fact that many different optimization techniques can potentially be used makes these approaches more appealing.

One of the best studied and well-known replacement approaches consists of the so-called *adaptive* algorithms, in which prototype features are considered as weights in some sort of connectionist system. Then, a criterion function is defined which leads to a reward-punishment modification rule which permits the update of prototype locations over a convenient number of iterations. Because of its simplicity and surprisingly excellent results in practice, the LVQ approach introduced by Kohonen, based on his self organizing maps [7], [10], has sometimes eclipsed other interesting approaches based on the same idea [11], [12].

An alternative to adaptive algorithms consists of the so-called *agglomerative approach* in which groups of samples are replaced by a representative in different iterations driven by a convenient criterion related to the discriminating power of the obtained prototypes. The main advantage of these algorithms with regard to the adaptive ones is that they are unaffected by initialization and that the final number of prototypes is decided by the algorithm itself and need not be prespecified. Chang [6] proposed one of the first condensing methods based on this strategy. It begins with a training set T, considering all the samples in T as initial prototypes. The algorithm iteratively attempts to merge the two closest prototypes (p, q) of the same class and replace them with their weighted average prototype p^*, until no new merge is possible. A merge is accepted if the new set of prototypes does not misclassify any pattern in T. This property is called *consistency* and is shared by a number of both replacement and selection algorithms. In the particular case of LVQ, this property is not considered and, consequently, the sets of prototypes obtained are not necessarily consistent.

The *modified Chang algorithm (MCA)* presented by Bezdek *et al.* [8] constitutes a slight improvement over the previous algorithm. With regard to the original algorithm of Chang, the MCA uses the simple arithmetic mean between p and q to compute p^* instead of the weighted mean. This scheme also implies an algorithmic change in the way in which pairs of prototypes are considered, introducing a different way of merging. As a consequence, the MCA generally obtains smaller sets of prototypes than Chang's algorithm. A computational improvement is also achieved based on storing cross distances among prototypes in the same class only. Nevertheless, the strategy behind the original idea remains unchanged.

Both algorithms have two kinds of drawbacks. First, they use a restricted strategy for building prototypes based on pairwise merging only, and consequently, they may provide condensing results which are far from the optimal ones, both from the point of view of their size and their representativity. Second, they employ a considerable amount of computation to exhaustively check consistency for any possible merging.

III. A New Generalized Prototype Merging Strategy

A further generalization of the idea of merging prototypes while maintaining consistency has been recently proposed [9]. From an algorithmic point of view, our proposal is quite similar to a classical hierarchical agglomerative clustering [13], in which pairs of clusters are successively merged according to a minimum intercluster criterion. The difference is that now merging occurs only between clusters with the same class label and only if the consistency of the resulting cluster representatives with regard to the whole input set is guaranteed.

This algorithmic scheme extends the concept of prototype by attaching a subset of samples which are close enough to it. The main idea consists of considering clusters of initially given (labeled) samples and their cluster representatives as some sort of extended prototypes. In this way, the merging of prototypes becomes a union of two clusters of the same class (and the recomputation of the new representative), while the distance between prototypes becomes a cluster distance.

The use of clusters leads to a merging process where prototypes are built on the basis of distances between the whole clusters (neighborhood) that they represent, which tends to result in a more meaningful placement of the final prototypes. Additionally, clusters lead to important computational shortcuts when checking the consistency of prototypes.

The fact that each cluster representative is responsible for the correct classification of all its members will be called *cluster consistency*. More formally, a set of (labeled) prototypes P is said to be *prototype-consistent* with respect to a set of initially given prototypes T, if every point in T is correctly classified by the 1-NN rule using P as a reference set. On the other hand, a partition of a set of initially given prototypes T into clusters (with the corresponding set of cluster representatives, P) is said to be *cluster-consistent* if every point in T is closer to its representative than to any other prototype in P with a different class label.

It trivially follows from these definitions that cluster consistency implies prototype consistency. In general, the converse is not true, but cluster consistency can always be easily recovered given a prototype-consistent set by simply reassigning each point to the cluster representative which is the closest (which is from the same class by the consistency property).

IV. Efficient Consistency Verification Procedure

If the consistency of p prototypes with regard to n original ones is checked exhaustively [6], [8], it requires $\mathcal{O}(np)$ time which makes any

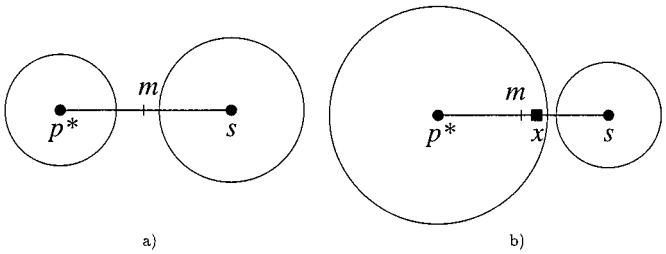

Fig. 1. Geometric representation of (2) which is sufficient to test cluster consistency: (a) the condition is satisfied and (b) the condition is not satisfied (a sample x could exist which is represented by p^* but which is closer to s).

agglomerative consistency-based approach to run at least in $\mathcal{O}(n^3)$. Moreover, the empirically observed hidden constants in this asymptotic result appear to be quite large.

A cluster-based consistency checking procedure could benefit from early consistency confirmation, by using the concept of cluster consistency and some geometric properties of the clusters. The main idea of this new consistency verification procedure can be detailed as follows. Given a cluster-consistent partition of a set of labeled prototypes (T and the associated set of representatives P), the partition resulting from combining two clusters into a new one which is represented by p^* from class k^* will also be cluster-consistent if $\forall s \in P$, the class of which is different from k^*

$$\begin{aligned} d(s,x) &> d(p^*,x)\,,\ \forall x \text{ in the cluster of } p^* \\ d(s,y) &< d(p^*,y)\,,\ \forall y \text{ in the cluster of } s. \end{aligned} \tag{1}$$

This fact can be easily proved. It is worth noting that checking cluster consistency using (1) and consequently, the induced prototype consistency, does not improve the exhaustive procedure of the consistency checking mentioned above. The assumption of some properties on the distance function and the use of simple geometric information of the clusters, leads to some shortcuts in checking consistency.

Let r_p be the radius associated with the cluster represented by prototype p, in such a way that any sample in this cluster is at a distance from p which is not greater than r_p. Given a cluster-consistent partition of T and the associated set of representatives P, the partition resulting from combining two clusters into a new one (which is represented by p^* from class k^*) will satisfy (1) if

$$\forall s \in P : \text{class}(s) \neq k^*,\ d(p^*,s) > 2 \cdot \max(r_{p^*}, r_s)\,. \tag{2}$$

The fulfillment of the *triangular inequality* and *symmetry* on the distance function, leads to a straightforward proof of the previous fact. As a consequence, (2) is sufficient for cluster consistency. Fig. 1 shows a simple example where (2) is used to assess cluster consistency.

A sufficient condition for (2) can be obtained by considering only the closest representatives for each class and the maximum radius in the corresponding class. Given a cluster-consistent partition of T and the associated set of representatives P, the partition resulting from combining two clusters into a new one (which is represented by p^* from class k^*) will satisfy (2) if

$$\forall k \neq k^*,\ d\left(p^*, s^k\right) > 2 \cdot \max\left(r_{p^*}, r^k\right) \tag{3}$$

where s^k is the prototype which is closest to p^* and r^k is the radius of the maximum-radius cluster in class k.

Note that the rest of the cluster representatives of each class $k \neq k^*$ are located farther from p^* than s^k and their associated radii are no larger than r^k. From the previous analysis, it can be deduced that (3) is sufficient for cluster consistency.

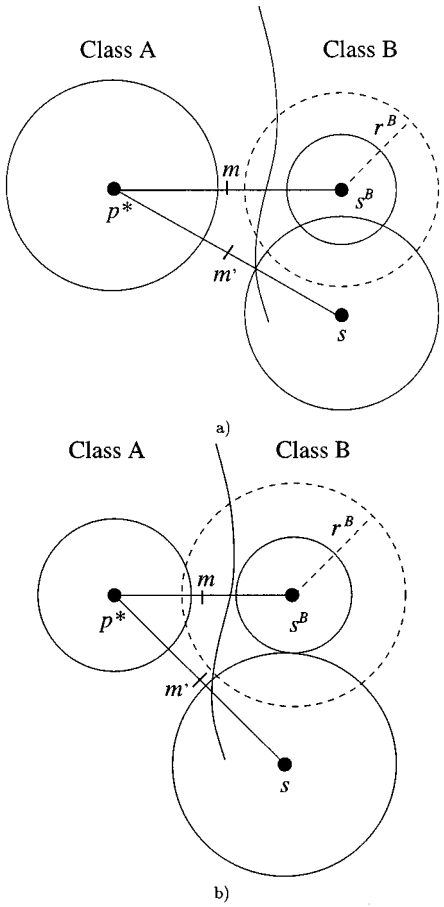

Fig. 2. Illustrative examples. Let m and m' be the midpoints between prototypes. In (a), consistency can be guaranteed by (3). No sample can possibly create an inconsistency because other class B representatives must be farther than s^B and with smaller or equal radius than r^B. In (b), (2) is needed to guarantee consistency because (3) is not fulfilled. All clusters in class B (the ones represented by s^B and s) need to be visited.

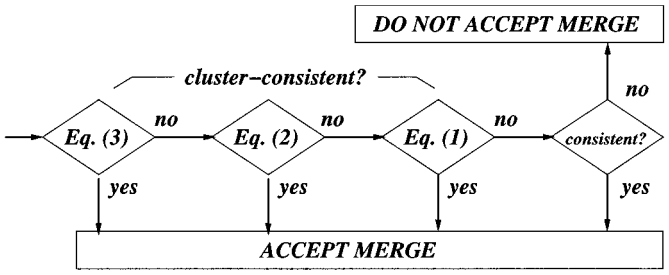

Fig. 3. Schematic description of the consistency checking procedure.

Fig. 2(a) illustrates a situation in which cluster consistency can be assessed using (3). A different one is shown in Fig. 2(b) in which (3) fails and (2) confirms the cluster consistency.

Condition (3) can be used as a *first* stage, (2) can be used as a *second* stage, and (1) can be used as a *third* stage of an efficient scheme to check cluster consistency and consequently, prototype consistency (the final goal). These three stages can be used in a complementary way (see Fig. 3). When (3) is unable to assess cluster consistency for a specific class k, (2) attempts to check it for each individual cluster in class k. If the latter fails for some cluster s member of class k, then (1) is used for each sample y which is a member of the cluster represented by s and also for each sample x member of p^* (new agglomerated cluster). In case of failure of (1) for any sample x or y, the current local structure is

not cluster-consistent and a direct verification of prototype consistency is needed.

In this context, a so-called *fourth* stage is used for checking prototype consistency. It requires looking for the same-class nearest prototype to x or y (which does not need to be their corresponding prototype or cluster representative), which is producing the cluster inconsistency. The distance between them is compared with the distance between the sample and the other-class prototype involved. If the sample is correctly classified, it must be moved from its current cluster to the cluster represented by its same-class nearest prototype, transforming the current prototype consistency into cluster consistency.

The computational cost of the four stages of the previous algorithm are $\mathcal{O}(p)$, $\mathcal{O}(p)$, $\mathcal{O}(np)$, and $\mathcal{O}(np)$, respectively. Nevertheless, it has been observed that the third and fourth stages have a reduced influence in the work done to check consistency, especially for moderately-sized data sets. Even though the worst case complexity of the consistency checking procedure presented here is certainly in $\mathcal{O}(np)$, the empirically observed behavior of the final algorithm is significantly better than our MCA implementation.

The resulting method has been named *generalized modified Chang algorithm (GMCA)* [9]. In the particular case where the intercluster distance is the median [13], (which implies the simple mean as the way to agglomerate prototypes), GMCA results in an improved version of the MCA yielding identical results, but cutting the computing time by more than half in most cases. When other intercluster measures are used, smaller and better sets of prototypes (in the sense of classification power) can be built, reducing the condensing time even more.

V. Experimental Results

Several different groups of experiments have been performed to properly assess the merits and possible drawbacks of the presented techniques. In this paper, the goal consists of showing that GMCA clearly improves previously proposed merge-based condensing algorithms (which use consistency-like criteria) both in number of prototypes retained and representativity. In particular, experiments were conducted to compare GMCA (with different intercluster measures) and MCA with respect to the number of prototypes built, the error rate of the corresponding condensed 1-NN classification rule and the computation time. Also important from a practical point of view, the experimentation aims at showing that GMCA (possibly combined with editing) constitutes a valid alternative to the well-known and powerful family of LVQ-like algorithms.

The Euclidean distance has been taken as the generic distance between pairs of patterns on which the presented method is built. Four basic intercluster distances [13] were used: 1) average link (AV); 2) complete link (CO); 3) median (ME); and 4) the ward minimum variance method (WA). Also, the radius of the next agglomerated cluster[1] was used as an additional dissimilarity measure (RA). Following the suggestions found in [8], the Euclidean distance and the simple mean were always used as the procedure to compute new prototypes, except for Ward's method where the weighted mean and the squared Euclidean distance were considered because of its original formulation [13]. In this way, a proper comparison between GMCA and MCA results can be accomplished.

Both schemes (MCA and GMCA) were implemented in *C* programming language using the same data structures and code to make them as similar as possible. The major functional difference was in the checking-consistency procedure. The experiments were performed on a 450-MHz Intel Pentium II.

Four data sets were used to assess the behavior of the different merging schemes in several ways. These sets are the well-known Anderson Iris data [14], a synthetic two-dimensional (2-D) data set [15], the DNA data set [16], and the Landsat satellite image data [16], which are publicly available at UCI Machine Learning Repository. No normalization has been applied to these data sets in any case.

[1]That is, the distance between two clusters equals the radius of the cluster which results from their merge.

TABLE I
Condensing Results on the Iris Data Set. PTMCA and PDMCA Represent the Percentage of Time Spent and Distances Computed by GMCA With Regard to MCA, Respectively

Merging Scheme	No. of Prot.	PTMCA (%)	PDMCA (%)
MCA	11	100	100
GMCA+ME	11	44.44	28.53
GMCA+CO	10	66.67	55.43
GMCA+WA	9	66.67	36.34
GMCA+RA	10	66.67	45.56
GMCA+AV	11	66.67	61.92

The algorithms proposed need to be applied in practice to overlap-free data sets, as discussed in Section I. Nevertheless, the first data set is treated in a special way as in previous published works [6], [8], in order to obtain a zero resubstitution error rate starting from the whole data set.

As a general rule for the remaining data sets and according to previously published results [17], the Wilson editing [18] has been considered as a good compromise in practice to properly remove overlapping between classes. The only parameter involved (k) has been obtained in our experiments (unless otherwise stated) by performing a fivefold cross validation experiment using the training set *only* and computing the average classification accuracies for different values of k and comparing them to the "no editing" option. The best edited set (including the "no editing" option) is thus selected as input for the different GMCA algorithms [9].

The sets of prototypes obtained in this way have been compared in terms of size and error rate (using independent test sets) to the results obtained for LVQ1 and LVQ2 classifiers from the publicly available LVQ_PAK software [19]. Five different random initializations (ten in the case of synthetic data) have been considered using a prespecified number of codebook vectors in the range of the ones obtained through the GMCA approach. LVQ parameter setting and codebook initialization has been done exactly as suggested in [19]. Note that LVQ needs to be applied in general to unedited sets because the reward-punishment policy uses all the information about the data everywhere. In contrast, consistency-based methods implicitly use the information present in the margin between classes only.

A. Experiments on the Iris Data Set

This set has three classes that represent three varieties of iris flowers, namely, Setosa, Versicolor, and Virginica. Fifty samples from each class are available. Every sample is described by four measurements. Using the 150 prototypes available, Chang [6] and Bezdek *et al.* [8] reported two consistent sets of 14 and 11 prototypes, respectively, built by their schemes. The results obtained by GMCA are shown in Table I. The best result obtained with GMCA + WA consisted of only nine consistent prototypes. It is worth noting that all GMCA schemes were about twice as fast as MCA at obtaining solutions which were equal or better.

B. Experiments With Synthetic Data

The data set consists of two concentric bivariate normal distributions. Class 0 has zero mean and standard deviation equal to 1 in both dimensions, and class 1 has the same mean but standard deviation equal to 2 in both dimensions [15]. A training and test set consisting of 2500

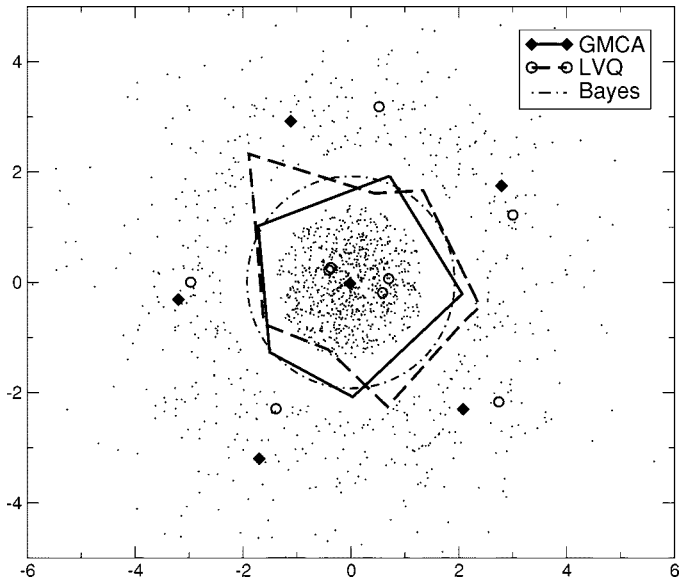

Fig. 4. Prototypes obtained on a particular run by LVQ2 (eight prespecified codevectors) and GMCA + WA (which obtains six prototypes) along with their corresponding decision boundaries for the 2-D concentric gaussians data set. The edited training set used by GMCA (dots) and the Bayes decision boundary are also shown as a reference.

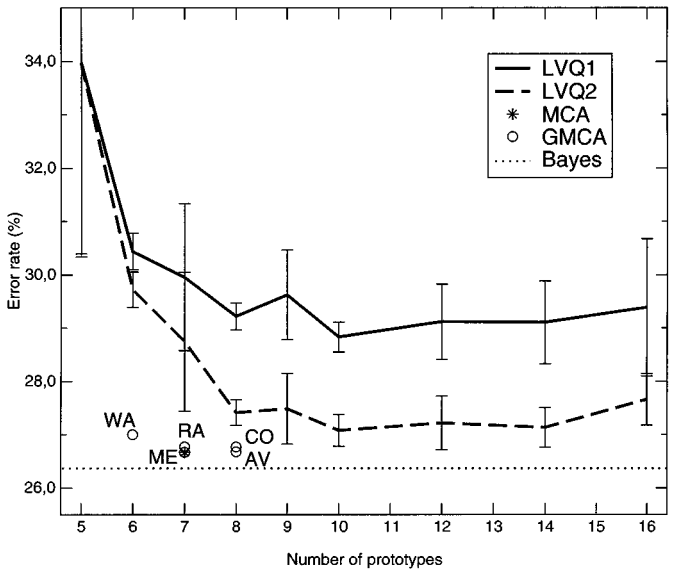

Fig. 5. Error rates and the corresponding standard deviations for different numbers of codevectors obtained with LVQ1 and LVQ2 algorithms, along with MCA and GMCA (with the intercluster dissimilarities: ME, CO, WA, RA, an AV) results using the 2-D concentric Gaussian data set. The Bayes error is also shown as a reference.

and 10 000 cases, respectively, have been randomly and independently generated for training and testing, respectively.

In order to isolate the effect of the particular editing on the final solution, an *optimal* Wilson editing has been used for this data set only. To obtain this, the Bayes rule with reject threshold $t = 0.40$ [20, p. 10] has been used as the internal classifier in the Wilson procedure. The particular reject threshold used was selected (based on the training set only) in order to obtain optimally (but realistic) edited sets of similar size and margin between classes compared to practical editing algorithms such as Wilson and Multiedit [2]. Typical sets of prototypes obtained in this way by GMCA (using Ward's distance in this case) and LVQ (LVQ2 using the whole training set) are shown in Fig. 4.

Ten different random initializations of LVQ algorithms with a number of codebook vectors ranging from 5 to 16 have been performed. The averaged error rates obtained are shown in Fig. 5 along with the corresponding number of prototypes. The Bayes error rate for this problem is 26.37 and is also shown.

The main fact that can be emphasized from this experiment is that all the methods obtain solutions relatively close to the optimal one but GMCA manages to represent the central cluster in an optimal way and consequently needs less prototypes to obtain similar or better results. It is worth noting that for this problem, very good results are also obtained if a different editing algorithm is used [9]. In particular, using Multiedit or Wilson editing with large values of k lead to results very similar to the ones shown in Fig. 5.

C. Experiments on the DNA Data Set

This data set corresponds to primate gene sequences. The problem is to recognize boundaries between different parts of the DNA. There are two sets, one for training composed by 2000 samples and one for testing with 1186 samples, which are partitioned into three classes. Each sample is described by 180 binary attributes (Statlog version) [16].

In all GMCA schemes, error rates were estimated on the test set by using the 1-NN rule with the condensed sets of prototypes which were built from the *unedited* training set, which was the best among all editing options according to above established procedure. Table II

TABLE II
CONDENSING RESULTS ON THE DNA DATA SET. PTMCA AND PDMCA REPRESENT THE PERCENTAGE OF TIME SPENT AND DISTANCES COMPUTED BY GMCA WITH REGARD TO MCA, RESPECTIVELY

Merging Scheme	No. of Prot.	Error Rate(%)	PTMCA (%)	PDMCA (%)
MCA	193	21.33	100	100
GMCA+ME	193	21.33	29.71	29.70
GMCA+CO	68	12.48	0.52	0.16
GMCA+WA	59	10.12	1.22	0.10
GMCA+RA	65	11.38	2.97	0.09
GMCA+AV	63	11.89	1.40	0.49

lists the condensing results. The classification error rates on the test set using the 1-NN and the 30-NN (best k-NN) rules with the original training set were 23.44% and 13.07%, respectively.

GMCA merging schemes yielded smaller sets of prototypes than MCA, which were much better at classifying the test set. At the same time, GMCA schemes achieved a dramatic reduction in the resources (condensing time and computed distances) required by MCA.

Error rates and the number of prototypes of these approaches (the same listed in Table II) along with the results obtained with LVQ algorithms are shown in Fig. 6. As could be expected, the LVQ2 was slightly (but clearly) better than all the other algorithms (including LVQ1) which exhibited very similar error rates (apart from MCA). This behavior can be related to the fact that GMCA is constrained to obtain consistent sets which may lead to some inaccuracies in the regions close to class boundaries. Conversely, it is well known that LVQ approaches are able to deal with very high overlapping, provided a good initialization has been done. The fact that GMCA results lie between LVQ1 and LVQ2 can be considered as a good result for this high-dimensional problem.

D. Experiments on the Landsat Satellite Image Data

This database consists of the multispectral values of pixels in 3×3 neighborhoods in a satellite image and the classification associated

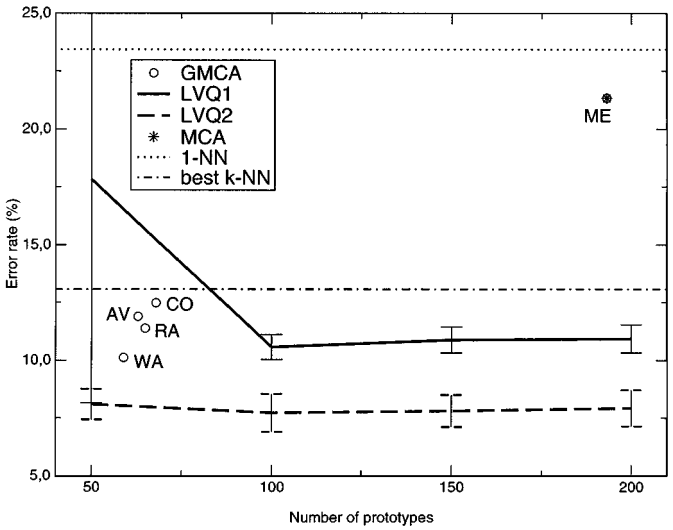

Fig. 6. Error rates and the corresponding standard deviations for different numbers of codevectors obtained with LVQ1 and LVQ2 algorithms, along with MCA and GMCA (with the intercluster dissimilarities: ME, CO, WA, RA, and AV) results using the DNA data set.

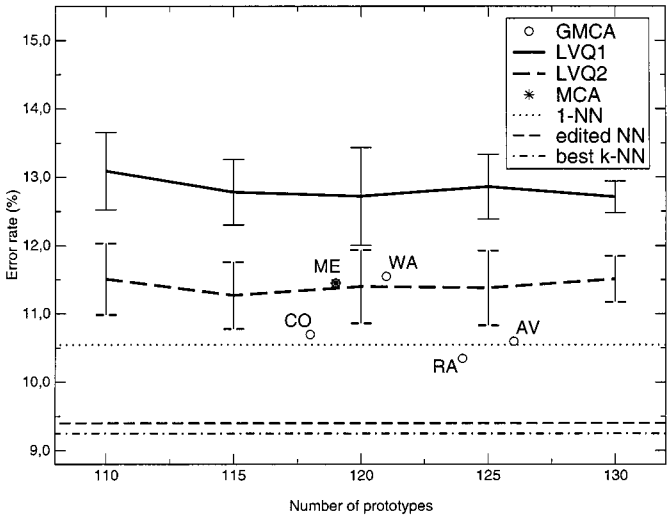

Fig. 7. Error rates and the corresponding standard deviations for different numbers of codevectors obtained with LVQ1 and LVQ2 algorithms, along with MCA and GMCA (with the intercluster dissimilarities: ME, CO, WA, RA, and AV) results using the Landsat satellite image data set.

TABLE III
CONDENSING RESULTS ON THE LANDSAT SATELLITE IMAGE DATA SET. PTMCA AND PDMCA REPRESENT THE PERCENTAGE OF TIME SPENT AND DISTANCES COMPUTED BY GMCA WITH REGARD TO MCA, RESPECTIVELY

Merging Scheme	No. of Prot.	Error Rate(%)	PTMCA (%)	PDMCA (%)
MCA	119	11.45	100	100
GMCA+ME	119	11.45	42.02	34.32
GMCA+CO	118	10.70	16.91	10.15
GMCA+WA	121	11.55	29.24	6.80
GMCA+RA	124	10.35	26.72	9.18
GMCA+AV	126	10.60	29.63	21.37

with the central pixel in each neighborhood [16]. The aim is to identify regions with different soils and crops. There are two sets, one for training with 4435 samples and one for testing with 2000 samples, which are partitioned into six classes of 36-dimensional data.

The edited training set with $k = 8$ (4018 samples) was selected in this case as the input for the different merging schemes. The error rates were estimated on the test set by using the 1-NN rule with the condensed sets of prototypes. Table III lists the condensing results on the edited training set including the estimated error rates. As a reference, the classification results on the test set using the 1-NN, the *edited* 1-NN, and the 4-NN (best k-NN classifier) rules with the original training set were 10.55%, 9.40%, and 9.25%, respectively.

The same results (error rates and number of prototypes) along with the ones obtained with the LVQ approach using the original (unedited) set are shown in Fig. 7. In this experiment, less significant differences with regard to the final number of prototypes is obtained. Nevertheless, it is worth noting that the GMCA approaches generally give better results in performance to those of the LVQ classifiers.

With the only exception of Ward's method, GMCA merging schemes were superior to MCA, achieving better sets of prototypes and remarkably faster speed than MCA. This suggests that GMCA approaches using distances CO, RA, and AV lead to clusters which adapt to the data in a better way. The fact that GMCA was generally superior to LVQ (with regard to error rates obtained) using a limited number of prototypes, can be considered as a very good result for this experiment, even though these results stem from the combination of editing and GMCA.

VI. CONCLUSIONS AND FURTHER WORK

A generalized condensing scheme based on class-conditional hierarchical clustering (GMCA) is proposed. The basic idea is to replace a group of prototypes by a representative while keeping the consistency property. The algorithm improves and generalizes previous works by explicitly introducing the concept of cluster and cluster consistency. The use of geometric cluster properties produces an efficient merging scheme based on local consistency verification, guaranteeing the entire system consistency while minimizing the computation needed.

MCA was experimentally compared with merging schemes induced by GMCA taking into account five different intercluster dissimilarity measures. In the particular case of the GMCA with the median intercluster distance (GMCA + ME), which yields identical sets of prototypes as MCA, a notable reduction in the time and the computed distances required was achieved in all experiments. When other intercluster measures were used, smaller and better sets of prototypes (in the sense of classification power) were built, reducing the condensing time even more.

The GMCA approach is able to obtain results which are similar to or even better than the well-known LVQ approach which is supposed to obtain close-to-optimal results when initialized properly. This suggests that GMCA is able to adapt to the underlying distribution of the data while significantly maintaining discriminating power. An obvious extension of the present work would consist of relaxing the consistency requirement and obtaining a hybrid approach by combining the "agglomerative" strategy of GMCA and the "adaptive" one from LVQ.

REFERENCES

[1] B. V. Dasarathy, *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques*, B. V. Dasarathy, Ed: IEEE Comput. Soc. Press, 1991.
[2] P. Devijver and J. Kittler, *Pattern Recognition: A Statistical Approach.* Englewood Cliffs, NJ: Prentince-Hall, 1982.
[3] P. E. Hart, "The condensed nearest neighbor rule," *IEEE Trans. Inform. Theory*, vol. IT-14, pp. 515–516, May 1968.
[4] B. V. Dasarathy, "Minimal consistent set (MCS) identification for optimal nearest neighbor decision systems design," *IEEE Trans. Syst., Man, Cybern.*, vol. 24, pp. 511–517, Mar. 1994.
[5] L. I. Kuncheva and J. C. Bezdek, "Nearest prototype classification: Clustering, genetic algorithms, or random search?," *IEEE Trans. Syst., Man, Cybern. B*, vol. 28, pp. 160–164, Feb. 1998.

[6] C. L. Chang, "Finding prototypes for nearest neighbor classifiers," *IEEE Trans. Comput.*, vol. C-23, pp. 1179–1184, Nov. 1974.
[7] T. Kohonen, "The self-organizing map," *Proc. IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.
[8] J. C. Bezdek, T. R. Reichherzer, G. S. Lim, and Y. Attikiouzel, "Multiple-prototype classifier design," *IEEE Trans. Syst., Man, Cybern. B*, vol. 28, pp. 67–79, Feb. 1998.
[9] R. A. Mollineda, "Hierarchical agglomerative clustering for prototype selection and distance-based classification. use in cyclic strings," Ph.D. dissertation, Sch. Comput. Sci., Univ. Politècnica de València, València, Spain, 2001.
[10] T. Kohonen, *Self-Organizing Maps*. New York: Springer-Verlag, 1995.
[11] H. Yan, "Prototype optimization for nearest neighbor classifiers using a two-layer perceptron," *Pattern Recognit.*, vol. 26, no. 2, pp. 317–324, 1993.
[12] Y. S. Huang, K. Liu, and C. Y. Suen, "A new method of optimizing prototypes for nearest neighbor classifiers using a multilayer network," *Pattern Recognit. Lett.*, vol. 16, no. 1, pp. 77–82, 1995.
[13] F. Murtagh, "A survey of recent advances in hierarchical clustering algorithms," *Comput. J.*, vol. 26, no. 4, pp. 354–359, 1983.
[14] E. Anderson, "The IRISes of the Gaspe peninsula," in *Proc. Bull. Amer. IRIS Soc.*, vol. 59, 1935, pp. 2–5.
[15] T. Kohonen, "Statistical pattern recognition with neural networks: Benchmarking studies," in *Proc. IJCNN*, vol. I, San Diego, CA, 1988, pp. 61–68.
[16] R. D. King, "Statlog Databases," Dept. Statist. Modeling Scie., Univ. of Strathclyde, Glasgow, U.K., 1992.
[17] J. S. Sánchez, F. Pla, and F. J. Ferri, "Prototype selection for the nearest neighbor rule through proximity graphs," *Pattern Recognit. Lett.*, vol. 18, no. 6, pp. 507–513, 1997.
[18] D. L. Wilson, "Asymptotic properties of nearest neighbor rules using edited data," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-2, pp. 408–421, 1972.
[19] (1997) LVQ PAK: The learning vector quantization package. Helsinki Univ. Technol., Helskinki, Finland. [Online]. Available: http://www.cis.hu.fi/research/lvq_pak
[20] A. Webb, *Statistical Pattern Recognition*. London, U.K.: E. Arnold, 1999.

A Reduction Approach for Fuzzy Rule Bases of Fuzzy Controllers

C. W. Tao

***Abstract*—In this paper, a new approach to reducing the number of rules in a given fuzzy rule base of a fuzzy controller is presented. The fuzzy mechanism of the fuzzy controller under consideration consists of the product-sum inference, singleton output consequents and centroid defuzzification. The output consequents in the cells of the rule table are collected and represented as an output consequent matrix. The feature of the output consequent matrix is extracted by the singular values of the matrix. The output consequent matrix is reasonably approximated with a dominant consequent matrix. Also, the elements of the dominant consequent matrix is determined to minimize the approximation error function. Then the size of the dominant consequent matrix (the size of the fuzzy rule base) is reduced through the rule combination approach. The scaling factors for the fuzzy controller with the reduced rule table are adjusted to have the control system satisfy the performance indexes. The effectiveness of the proposed approach is shown using simulation and experiment results.**

***Index Terms*—Fuzzy controllers, reduction of rule bases, singular value decomposition.**

I. Introduction

Fuzzy logic techniques [2], [14] implementing the experts' knowledge and experiences have been widely applied to many complex control systems with unknown dynamics [1], [10]. To apply the rules with linguistic predicates [4], [5], [7]–[9] from experts, a fuzzy mechanism is usually designed to have a rule base with fuzzy if–then rules [6]. Since, it is possible to include redundant rules, the complex fuzzy rule bases may suffer from the disadvantages like heavy computation load and large memory space. Thus, to refine the set of fuzzy rules to form a concise fuzzy rule base is a desirable process. Recently, Yam *et al.* [11] provided a singular value decomposition (SVD) approach for reduction of fuzzy rule bases. However, the number of active rules for each input of the fuzzy mechanism is increased due to the new membership functions generated from the reduction process [11]. The increase in the number of active rules makes the computation load heavier when the output of the fuzzy controller is calculated. Moreover, although the output error of the fuzzy controller is bounded [11], the performance of the system output can be further improved.

To make improvement from the work in [11], a new approach for reducing a given fuzzy rule base of a fuzzy controller is designed based on the idea of reduction of fuzzy rule base in [11]. As in Yam's paper [11], the output consequents in the cells of a fuzzy rule table are collected and represented as an output consequent matrix. The characteristics of the output consequent matrix are featured by the singular values of the matrix. Unlike the reduction approach in [11], the output consequent matrix is reasonably approximated with a dominant consequent matrix in this paper. The elements of the dominant consequent matrix are determined by minimizing the approximation error function. With the rule combination approach, the size of the dominant output consequent matrix is reduced. Therefore, the size of the fuzzy rule base is reduced.

Manuscript received August 17, 2000; revised December 19, 2001. This work was supported in part by the National Science Council, Taiwan, R.O.C., under Grant NSC 90-2213-E-197-002. This paper was recommended by Associate Editor W. Pedrycz.

The author is with the Department of Electrical Engineering, National I-Lan Institute of Technology, I-Lan, Taiwan, R.O.C. (e-mail: cwtao@mail.ilantech.edu.tw).

Publisher Item Identifier S 1083-4419(02)05725-4.