Continuous Optimization Schemes for Fuzzy Classification

K. Blekas, G. Papageorgiou and A. Stafylopatis
National Technical University of Athens
Department of Electrical and Computer Engineering
157 73 Zographou, Athens, Greece
email: {kblekas,gpap,andreas}@softlab.ece.ntua.gr

Abstract

Two approaches are developed, which are suitable for the optimization of a fuzzy classification scheme through the formation of appropriate space-filling clusters. The first approach is based on the analog Hopfield neural network, while the second one uses real-encoded genetic optimization. Experimental results concerning difficult classification problems show that both proposed approaches are very successful in generating fuzzy partitions and outperform other known algorithms in terms of the correct placement of patterns into partitions.

1 Introduction

The task of pattern classification is a key element to numerous engineering applications and typically constitutes a major component of an intelligent diagnostic system. Unlike pattern clustering, which considers a set of unlabeled data objects and seeks to find natural groupings amongst the exemplars, pattern classification provides class labels with pattern exemplars and attempts to find the decision boundary between classes that minimizes misclassification. We shall consider here a nearest-neighbour pattern classification scheme that is able to build decision regions separating classes of any shape and size. Class regions are created by properly placing and adjusting a number of fuzzy clusters, characterized by their respective geometric centres. The clusters (whose number is assumed to be known a priori) are formed according to a predefined criterion, based on a distance measure and on the knowledge of pattern labels.

A broad spectrum of clustering/classification algorithms attempt to generate a partition of the sample data through the minimization of an objective function based on the clustering criterion. The kinds of partitions generated and the geometric structure of the clusters are closely related to the distance measure chosen and the objective function being optimized. The partitions are either hard, that is each sample point is unequivocally assigned to a cluster and is considered to bear no similarity to members of other clusters, or fuzzy, in which case a membership function expresses the degree of similarity between the sample and each cluster [1, 2, 6, 15, 16].

As an alternative to typical numerical processes used to search for an optimal partitioning, computationally "intelligent" techniques such as neural networks and genetic algorithms have recently been applied to related problems [3, 4, 5, 10, 11, 17]. In the present paper we develop two approaches of the above type, namely a scheme based on the analog Hopfield network and a real-coded genetic algorithm, to obtain fuzzy partitioning optimization, and evaluate their effectiveness on various data sets.

Let $X = \{\vec{x}_1, \ldots, \vec{x}_n\}$ denote a set of *labeled* data points (patterns) in $\mathcal{R}^p$. Each $\vec{x}_j$ is the numerical representation of p features associated with a corresponding physical object. Our aim is to partition the p-dimensional space into a number c of clusters ($1 < c < n$), that is assumed to be known in advance. Clusters are represented by the vector $V = [\vec{v}_1, \ldots, \vec{v}_c]$ ($\vec{v}_i \in \mathcal{R}^p$) of their geometric centres (cluster prototypes). We shall consider that each cluster belongs to one of the predefined classes, namely the classes of pattern labels. We seek the best set of clusters, such that patterns are assigned to clusters according to a fuzzy membership function with minimum misclassification error.

Given X, a partition of X is represented by the $c \times n$ fuzzy partition matrix $M = [\mu_{ij}]$ satisfying the conditions: $0 \leq \mu_{ij} \leq 1$ ($1 \leq i \leq c$, $1 \leq j \leq n$), $\sum_{i=1}^{c} \mu_{ij} = 1$ ($1 \leq j \leq n$) and $\sum_{j=1}^{n} \mu_{ij} > 0$ ($1 \leq i \leq c$), where each value μ_{ij} represents the membership of the j-th data point to the i-th cluster.

The optimization criterion is associated with the generalized least-squared errors functional

$$H = \sum_{i=1}^{c} \sum_{j=1}^{n} \zeta_{ij} D_{ij} \mu_{ij}^{m} \tag{1}$$

where $m > 1$ is a weighting exponent (degree of fuzzification) and D_{ij} is some similarity metric between $\vec{x}_j$ and $\vec{v}_i$, which will be taken equal to the squared Euclidean norm:

$$D_{ij} = ||\vec{x}_j - \vec{v}_i||^2 \tag{2}$$

The quantity ζ_{ij} is equal to unity if the pattern $\vec{x}_j$ and the centre $\vec{v}_i$ are of the same class, otherwise it is taken equal to some positive constant much larger than unity.

Thus, the optimization criterion includes distance information as well as knowledge about pattern and cluster labels, in a way to favour correct classification. In general, the choice of the similarity metric implies different geometric and statistical properties of the generated partition.

2 Hopfield Network Approach

The analog Hopfield neural network [9] is a fully connected, continuous time network, that employs units with analog input and output. The basic idea is to encode the objective function and the problem constraints in terms of an appropriate energy function which can be minimized by the network architecture. An analog Hopfield neural network with N units performs local search in the continuous space inside the hybercube $\{0,1\}^N$. We consider a network with connection weights w_{ij}, where $w_{ii} = 0$ and $w_{ij} = w_{ji}$, and threshold values θ_i $(i, j = 1, \ldots, N)$. By u_i and y_i we denote the input and the output of unit i, respectively. The energy function:

$$E = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij} y_i y_j - \sum_{i=1}^{N} \theta_i y_i + \eta \sum_{i=1}^{N} \int_0^{y_i} f^{-1}(x)dx \quad (3)$$

constitutes a Liapunov function for the system. This function decreases during the operation of the network. The evolution of the behaviour of each unit is described by the following equations:

$$\frac{du_i}{dt} = -\frac{\partial E}{\partial y_i} = -\eta u_i + \sum_{j=1}^{N} w_{ij} y_j + \theta_i \quad (4)$$

and

$$y_i = f(u_i) \quad (5)$$

where f is a differentiable, monotonically increasing function with values in $[0, 1]$ (or in $[-1, 1]$). A standard form of f, which we are going to use, is $y_i = f(u_i) = \frac{1}{2}(1 + \tanh(u_i/u_0))$, where u_0 determines the steepness of the gain. Under the above dynamics, the analog Hopfield network converges to an equilibrium state that corresponds to a local minimum of the energy function.

A formulation of the problem suitable for the Hopfield network requires the definition of an appropriate quadratic energy function. The energy function will consist of three parts E_1, E_2 and E_3 (see equations (6)-(9)). The E_1 part consists of the constraint terms, which enforce the network to relax at states at which each pattern $\vec{x}_i$ is characterized by high membership to one cluster. The E_2 part includes the standard integral term, which guides the network to search in the interior of the unit hypercube $\{0,1\}^{n \cdot c}$, and the thresholds of the units of the network. Finally, the E_3 component consists of the cost term which is the functional H defined in Section 1. The coefficients A, B and C are positive constants.

$$E = E_1 + E_2 + E_3 \quad (6)$$

$$E_1 = \frac{A}{2}\sum_{j=1}^{n}\sum_{i=1}^{c}\sum_{\substack{k=1\\k\neq i}}^{c} \mu_{ij}\mu_{kj} + \frac{B}{2}\sum_{j=1}^{n}(\sum_{i=1}^{c}\mu_{ij} - 1)^2 \quad (7)$$

$$E_2 = \eta \sum_{i=1}^{c}\sum_{j=1}^{n} \int_0^{\mu_{ij}} f^{-1}(x)dx - \sum_{i=1}^{c}\sum_{j=1}^{n} \theta_{ij} \quad (8)$$

$$E_3 = C \cdot H = C \sum_{i=1}^{c}\sum_{j=1}^{n} \zeta_{ij} D_{ij} \mu_{ij}^m \quad (9)$$

The relaxation of the network is performed according to the following differential equations:

$$\frac{du_{ij}}{dt} = -\frac{\partial E}{\partial \mu_{ij}} = -A\sum_{\substack{k=1\\k\neq i}}^{c}\mu_{kj} - B(\sum_{k=1}^{c}\mu_{kj} - 1)$$

$$-\eta u_{ij} + \theta_{ij} - C\zeta_{ij}\frac{\partial D_{ij}}{\partial \mu_{ij}}\mu_{ij}^m - mC\zeta_{ij}D_{ij}\mu_{ij}^{m-1} \quad (10)$$

The above equations are solved using the Euler approximation method.

At each relaxation step the centres of the clusters are updated as follows:

$$\vec{v}_i = \left(\sum_{j=1}^{n}(\mu_{ij})^m \vec{x}_j\right) / \sum_{j=1}^{n}(\mu_{ij})^m \quad (11)$$

The distances of the patterns from the new centres are computed according to (2), while the derivative $\partial D_{ij}/\partial \mu_{ij}$ is obtained from (2) and (11). At each iteration, the u_{ij} provided by the Euler method yield the new μ_{ij} values.

3 Real-coded Genetic Optimization

Genetic algorithms [7, 13] seem to offer a promising alternative in the direction of optimizing the proposed functional H. As genetic algorithms are considered more natural for discrete optimization problems, most approaches to clustering have been based on a binary encoding of the parameter space [3, 5]. The traditional binary coding, however, has serious drawbacks when applied to multidimensional problems of high numerical precision, since binary representation generates prohibitively large search spaces. For problems with variables over continuous domains, it seems natural to represent genes directly as floating-point numbers and chromosomes as vectors of real numbers, thus enabling the exploration of large domains without sacrificing precision or memory [8, 13].

In the approach presented here, we have considered floating-point encoding of the cluster centres $\vec{v}_i = [v_{ik}]$ $(1 \le i \le c,\ 1 \le k \le p)$. A chromosome is a vector (string) of total length $c \times p$, composed of c subvectors of length p corresponding to cluster centres. Given a vector V, the membership values μ_{ij} $(1 \le i \le c, 1 \le j \le n)$ are eliminated from H through the substitution

$$\mu_{ij} = \left[\sum_{l=1}^{c}\frac{D_{ij}}{D_{lj}}\right]^{-1} \quad 1 \le i \le c,\ 1 \le j \le n \quad (12)$$

and the computed value of the function H provides the fitness of the string. (Actually, the fitness is obtained via appropriate normalization of H yielding a form suitable for maximization.)

The main features of the proposed real-coded genetic algorithm (RCGA) are described next. Given an initial population of K randomly created vectors, a reproduction procedure takes place repeatedly, during which members of the population are recombined and a new generation of vectors is created, until a maximum number of generations is attained. Specifically, at each generation step and for each current member α we decide with probability p_c (which generally assumes high values) whether crossover will be applied or not, and, in the positive case, another member β is randomly selected and crossover is performed between the vectors corresponding to the two parents. The recombination operator that we have considered is a variant of single-point crossover operating at the cluster level. A position s is selected at random within the range $1 \leq s \leq c-1$, where c is the number of subvectors, and crossover is performed between each pair of corresponding subvectors $s, \ldots, c$ of each vector of V^α and V^β The above reproduction scheme is *synchronous*, in the sense that all K children can be created simultaneously and independently based on the precedent generation, thus achieving a high degree of parallelism. In implementing the crossover scheme we have used selection based on fitness value, that is members with high fitness value have more chances to be selected and recombined in the next generation.

Population diversity is further enhanced by applying a mutation operator. In our case, we have considered an adaptation of the *non-uniform mutation* described in [13]. The non-uniform mutation operator can be applied to each individual real-valued component of each subvector of a vector with probability p_m, which generally takes very small values. Let us assume that the domain of the element v_{ik} ($1 \leq i \leq c$, $1 \leq k \leq p$) which will undergo mutation is $[a_{ik}, b_{ik}]$. If the operator is applied at generation step t and T is the maximum number of generations, then the new value of the element will be :

$$v'_{ik} = \begin{cases} v_{ik} + \Delta(t, b_{ik} - v_{ik}) & \text{if } \tau = 0 \\ v_{ik} - \Delta(t, v_{ik} - a_{ik}) & \text{if } \tau = 1 \end{cases} \tag{13}$$

with τ being a random binary digit (0 or 1), and

$$\Delta(t, y) = y\left(1 - r^{(1-\frac{t}{T})^b}\right) \tag{14}$$

where r is a random number from the interval $[0, 1]$ and b is a system parameter, which determines the degree of dependency on the number of generations. The function Δ returns a value in the range $[0, y]$ such that the probability of returning a number close to 0 increases as t increases. Thus, the size of the interval for the element value becomes lower with the passing of generations. This causes the mutation operator to make a uniform search of the space initially, and a very local search at later stages. It is this property that provides the system with fine local tuning capabilities.

A last operator which has proved to improve the described recombinative scheme is the *apathy* operator. When a vector of the current population gets a fitness value better than its previous one, it remains apathetic for a (generally small) number of generation steps. This technique does not cause any harm to the specific vector, while reinforcing other vectors by facilitating their participation in the selection process.

4 Experimental Results

In our experiments we have considered a variety of classification problems. The first data set that was used to test the proposed approaches was the Fisher's Iris database which consists of 150 feature vectors described by 4 continuous-valued attributes with 3 classes. Half of them were used for training and the remaining for testing (75 data). Another tested database was the James Cook University Thyroid gland database which is a collection of 215 instances whose 5 attributes are all continuous-valued. The vectors belong to one of three decision classes that define a prediction of a patient's thyroid to the class of euthyroidism, hypothyroidism or hyperthyroidism. We have selected 100 patterns for training and the remaining 115 for testing. Finally, we have used the sytnthetic two-class problem taken from Ripley [14]. It is a realistic problem with 1250 2-dimensional patterns that belong to 2 classes, where 250 and 1000 patterns were used for training and testing respectively.

Experiments have been conducted considering a range of values for the number c of clusters, applying the continuous Hopfield network and the real-coded genetic algorithm to each of the three data collections. In all experiments the output of the algorithms was the best solution found during the search (maximum 10000 iterations for the Hopfield network and 5000 genetic steps). After that, we were able to evaluate the classification rate of the optimized classifier as the percentage of patterns which are correctly placed into clusters (in the nearest-neighbour sense). It should be noted that both methods exhibited an almost perfect classification performance on the training sets.

For all datasets and for each number of clusters considered, a series of 10 experiments was performed using different seed values for the random number generator. For both optimization methods the weighting exponent m was set equal to 2. In the energy function E optimized by the Hopfield network the coefficients A, B and C were equal to 1.0, 1.0 and 5.0 respectively, whereas the thresholds θ_{ij} took the value 1.0 and the parameter η was set equal to 1.0. The population size of the genetic algorithm was nearly double the respective size of chromosomes. The crossover probability was $p_c = 0.85$, while the parameter b was set to 4.0. Finally, the mutation probability p_m had initially a large value (0.1) and was decreased by 3% every 100 generation steps until it reached the value of 0.005.

To evaluate the effectiveness of both approaches, the same experiments were carried out using a backpropagation feed-forward neural network (BP) and Kohonen's Learning Vector Quantization (LVQ) for the three databases. The neural network was a single hidden-layer network with 10 nodes, with a learning rate 0.09 and a momentum rate equal to 0.9. For the LVQ method we have used the LVQ1 algorithm [12] with learning rate 0.03.

Table 1 reports testing results for the three data collections and for the two proposed approaches in comparison with the BP and LVQ methods. The best number of clusters found and the classification rate for the testing sets are displayed. Also, the average number of steps required to find the best value is given for the Hopfield network and the genetic algorithm. The superiority of the proposed optimization techniques is apparent in all datasets. Using a small number of clusters the Hopfield network manages to relax at an equilibrium state where the corresponding centres, which were indirectly computed, show good classification behaviour. The genetic algorithm converged in a straightforward manner to a chromosome vector suggesting appropriate decision boundaries of pattern classes. As can be observed, the Hopfield network performs slighty better than the real coded-genetic algorithm. On the other hand the genetic algorithm needs little knowledge of the classification problem to perform efficient exploration of the domain space. It must be noted that for small numbers of clusters equal to the number of classes the results are very similar for both techniques.

	HOPF.	RCGA	BP	LVQ
	Iris database			
Clusters	3	6		6
Rate (%)	98.67	96.00	94.67	94.67
Avg. steps	4800	1950		
	Thyroid database			
Clusters	6	6		6
Rate (%)	96.52	96.52	94.78	94.78
Avg. steps	3780	2300		
	Synthetic database			
Clusters	4	4		6
Rate (%)	91.00	90.50	88.00	89.50
Avg. steps	4130	1570		

Table 1: Comparative results for the classification problems

5 Conclusions

We have developed and tested two approaches for the optimization of fuzzy partitioning in nearest-neighbour classification problems. Both approaches are based on the formulation of an appropriate clustering criterion that incorporates a distance metric along with knowledge about the class labeling of patterns and clusters. The first approach considers the use of an analog Hopfield network properly designed to encode the optimization criterion and problem constraints. The second approach is a genetic algorithm based on coding the space of cluster prototypes by means of floating point values and using appropriate genetic operators. The two methods have been tested experimentally on a variety of classification problems and have shown very good performance in terms of the rate of correct classifications during testing. Comparison with other established classification approaches has shown that the proposed formulation of the clustering criterion combined with the above optimization schemes provides a promising alternative. Moreover, this work allowed us to experiment with the use of continuous representations in a problem area that is particularly suited to this type of state-space encoding. The advantage of both schemes considered is reinforced by the possibility of obtaining efficient parallel implementations in a straightforward manner, thus ensuring fast and effective solutions to hard problems.

References

[1] Bezdek J.C., Ehrlich R. and Full W., FCM: The Fuzzy c-Means Clustering Algorithm. *Computers and Geosciences*, 10, 1984, pp. 191–203.

[2] Bezdek J.C. and Pal, S.K., *Fuzzy Models for Pattern Recognition*. IEEE Press, 1992

[3] Bezdek J.C, Boggavarapu S., Hall L.O. and Bensaid A., Genetic Algorithm Guided Clustering. *Proc. First IEEE Conf. on Evolutionary Computation*, Orlando, Florida, 1994, Vol. I, pp. 34–39.

[4] Blekas, K. and Stafylopatis, A., Real-coded Genetic Optimization of Fuzzy Clustering. *Proc. EUFIT'96*, Aachen, Germany, 1996, pp. 461–465.

[5] Buckles B.P., Petry F.E., Prabhu D., George R. and Srikanth R., Fuzzy Clustering with Genetic Search. *Proc. First IEEE Conf. on Evolutionary Computation*, Orlando, Florida, Vol. I, 1994, pp. 46–50.

[6] Buhmann, J. and Kühnel, H., Complexity Optimized Data Clustering by Competitive Neural Networks. *Neural Computation*, 5, 1993, pp. 75–88.

[7] Goldberg D.E., *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.

[8] Herrera F., Lozano M. and Verdegay J.L., Tackling Real-Coded Genetic Algorithms: Operators and Tools for Behavioural Analysis. *Technical Report #DECSAI-95107*, Universidad de Granada, Spain, Feb. 1995.

[9] Hopfield, J.J. and Tank, D.W., Neural Computation of Decisions in Optimization Problems. *Biological Cybernetics*, 52, 1985, pp. 141–152.

[10] Kamgar-Parsi, B., Gualtieri, J.A., Devaney, J.E. and Kamgar-Parsi, B., Clustering with Neural Networks. *Biological Cybernetics*, 63, 1990, 201–208.

[11] Kamgar-Parsi, B. and Kamgar-Parsi, B., A Revised Clustering Technique Using a Hopfield Network. *Proc. World Congress on Neural Networks*, Portland, Oregon, Vo. IV, 1993, pp. 24–27.

[12] Kohonen, T., The Self-Organizing Map. *Proceedings of the IEEE*, 78, 1990, 1464–1480.

[13] Michalewicz Z., *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, 1992.

[14] Ripley B.D., *Pattern Recognition and Neural Networks*, Cambridge University Press, 1996.

[15] Simpson P.K., Fuzzy Min-Max Neural Networks-Part 1: Classification. *IEEE Trans. on Neural Networks*, 3, 1992, 776–786.

[16] Simpson P.K., Fuzzy Min-Max Neural Networks-Part 2: Clustering. *IEEE Trans. on Fuzzy Systems*, 1, 1993, 32–45.

[17] Van Le T., Evolutionary Fuzzy Clustering. *Proc. IEEE Int. Conf. on Evolutionary Computation*, Perth, Western Australia, Vol. 2, 1995, pp. 753–758.