A Novel Approach to Unsupervised Robust Clustering using Genetic Niching

Olfa Nasraoui
Department of Electrical Engineering
University of Memphis
Memphis, TN 38152
onasraou@memphis.edu

Raghu Krishnapuram
Department of Mathematical and Computer Sciences
Colorado School of Mines
Golden, CO 80401
rkrishna@mines.edu

Abstract—
We present a new unsupervised robust clustering algorithm that can successfully find dense areas (clusters) in feature space and determine their number. The clustering problem is converted to a multimodal function optimization problem within the context of Genetic Niching. The niche peaks, which constitute the final cluster centers, are identified based on Deterministic Crowding (DC). The problem of crossover interactions in DC is eliminated by restricting the mating to members of the same niche only. Finally, the correct number of niche peaks or cluster centers is extracted from the final population. Genetic Optimization makes our approach much less prone to suboptimal solutions than other objective function based approaches, and frees it from the necessity of an analyitical derivation of the prototypes. As a result, our approach can handle a vast array of general subjective, even non-metric dissimilarities, and is thus useful in many applications such as Web and data mining. Additionally, the use of robust weights makes it less sensitive to the presence of noise than most traditional unsupervised clustering techniques.

I. Introduction

The Simple Genetic Algorithm (SGA) offers an efficient way to search for a global solution, and has been successfully used to search the solution space in clustering problems with a fixed number of clusters [1], and for robust clustering [2]. However, in practice, the number of clusters may not be known. When the number of clusters is unknown, the problem is sometimes called *unsupervised clustering*. Typically, unsupervised prototype-based clustering algorithms determine the correct number of clusters, C, using one of the following four approaches. The first approach repeats the clustering for several C values at a high computational cost, and uses a validity measure to choose the best partition. The second approach performs several passes through the data set, seeking and removing one cluster at a time [3]. The third approach consists of starting the clustering process with an overspecified number of clusters, and then merging similar clusters and eliminating spurious clusters [4]. The fourth and most recent approach is based on Competitive Agglomeration [5], in which an overspecified number of clusters compete against each other for data points, and clusters that lose in the competition gradually become depleted and vanish.

The difficulty in designing validity measures that can truly evaluate the goodness of a given cluster or partition is well known. Also, most validity measures either assume a known inlier/noise distribution or are very sensitive to noise. Another problem that plagues most clustering algorithms is their susceptibility to local minima. One way to avert this problem is to exploit the global optimization nature of SGA. Unfortunately, the genetic fitness measure needs to be based on validity, and most validity measures are monotonic with the number of clusters. This makes genetic optimization impossible. Also, because of the extreme selection pressure used in SGA, only the best individual (corresponding to the cluster with best fitness) can be expected to survive in the population.

In this paper, we propose a novel approach to unsupervised robust clustering using Genetic Algorithms (GA). We start by modifying our objective from searching the solution space for C clusters to searching this space for any one cluster. Accordingly, we need to optimize a fitness function that reaches a maximum at every good cluster center. Therefore, we need an optimization technique that can identify all the peaks in a multimodal fitness landscape. Because of the selection pressure in its reproduction step, the SGA is expected to locate only the highest peak. In order to locate all peaks, we resort to niching methods [6], [7], [8], [9] which can identify multiple optima within multimodal domains. We use Deterministic Crowding (DC) [9] as the genetic niching optimization tool. To alleviate the problem of crossover interaction between distinct niches, we propose an improved restricted mating scheme which

relies on an accurate and assumption-free estimate of the niche radii.

The remainder of this paper is organized as follows. In Section II, we give an overview of unsupervised clustering and genetic algorithms. In Section III, we present our new approach to unsupervised clustering based on genetic niching. In Section IV, we present our experimental results, and finally, we present our conclusions in Section V.

II. Background

A GA starts with an initial population of candidate solutions or individuals, and tries to modify them until the population converges to a solution. A problem-dependent fitness function must be chosen to measure the goodness of an individual. The modification of the population is done using an iterative application of selection, crossover, and mutation. SGA uses fitness-proportionate selection which is implemented using tournament or roulette wheel selection. Mating between two individuals is implemented using crossover which generally allows the creation of better children or offspring by combining the genetic materials of two parents with a large crossover probability P_c. These offspring will replace their parents in the next generation. Mutation is a totally random step, where each bit in the chromosome string of the offspring individuals is allowed to change value with a small mutation probability P_m.

The traditional GA has proved effective in exploring complicated fitness landscapes and converging populations of candidate solutions to a single global optimum. However, some optimization problems require the identification of global as well as local optima in a multimodal domain. As a result, several population diversity mechanisms have been proposed to counteract the convergence of the population to a single solution by maintaining a diverse population of members throughout its search.

De Jong [8] proposed Crowding methods which try to form and maintain niches by replacing population members preferrably with the most similar individuals. Unfortunately, stochastic "replacement errors" prevented this method from maintaining more than two peaks in a multimodal fitness landscape. Mahfoud [9] proposed an improved crowding mechanism, called "deterministic crowding" (DC), which nearly eliminated replacement errors and proved more effective in maintaining multiple niches. DC is presented below. In this algorithm, N_p is the number of individuals in the population, $f()$ is the fitness function, and $d()$ is a distance measure.

Deterministic Crowding (DC)

Repeat for G **generations** {
  Repeat $\frac{N_P}{2}$ **times** {
    Select two parents p_1 *and* p_2 *randomly without replacement;*
    Cross them to produce children c_1 *and* c_2*;*
    Optionally apply mutation to produce children c_1' *and* c_2'*;*
    IF $[d(p_1, c_1') + d(p_2, c_2')] \leq [d(p_1, c_2') + d(p_2, c_1')]$
    THEN {
      IF $f(c_1') > f(p_1)$ *THEN replace* p_1 *with* c_1'
      IF $f(c_2') > f(p_2)$ *THEN replace* p_2 *with* c_2'
    }
    ELSE {
      IF $f(c_2') > f(p_1)$ *THEN replace* p_1 *with* c_2'
      IF $f(c_1') > f(p_2)$ *THEN replace* p_2 *with* c_1'
    }
  }
}

Competition and improvement in DC occur only within the niches leading to a diverse population with members that are closer to the actual peaks. Unfortunately, DC suffers from crossover interactions between different niches. A dominated niche can, with another niche's assistance, cross to form members from a fitter (dominant) niche. This causes a migration of members from the dominated peaks to the dominant peaks that will only come to a halt when one of the dominated niches is depleted of its members.

III. New Approach to Unsupervised Robust Clustering using Genetic Niching

A. Representation and Initialization

The solution space for possible cluster centers consists of n-dimensional prototype vectors. These are represented by concatenating the Gray codes of the individual features for one cluster center into a binary string with 8 bits per feature value. Paradoxically, this means that the search space is much smaller than the one corresponding to using the SGA to search for C cluster centers as in the case of the Genetic C-LMedS [2]. In fact, if S is the search space size for our niching based unsupervised approach to clustering, then the search space size for the SGA based C-means clustering is S^C. As expected, the savings in the size of the search space translate into savings in the population size. The tremendous savings in terms of search space complexity are made possible by the advanced search process that will be described below. The initial centers are selected randomly from the set of feature vectors. This results in a population of N_P individuals, $P_{(i)}$, $i = 1, \cdots, N_P$.

B. Fitness Function

Since in general, we identify dense areas of a feature space as clusters, we will define the fitness value, f_i, for a candidate center location, $\mathbf{c}_i$, as the density of a hypothetical cluster at that location. For the case of 2-dimensional clusters, the density can be defined as

$$f_i = \frac{\sum_{j=1}^{N} w_{ij}}{\sigma_i^2}, \tag{1}$$

where $w_{ij} = \exp -\frac{d_{ij}^2}{2\sigma_i^2}$, σ_i^2 is a robust measure of scale (dispersion) for the i^{th} cluster, d_{ij}^2 is the Euclidean distance from data point $\mathbf{x}_j$ to cluster center $\mathbf{c}_i$, and N is the number of data points. It can easily be seen that as a variance measure, σ_i^2 is also related to the radius of the i^{th} niche, since in this particular optimization problem each cluster in the data set will generate a niche in the fitness landscape. More specifically, the niche radius is close to $K\sigma_i^2$, where K is approximately $\chi^2_{2,.995}$ for Gaussian clusters.

C. Mating Restriction

The DC selection and replacement procedure described in Section II will be used to advance from one generation to the next. However, some restrictions will be imposed on the mating of two parents to prevent different niches from interacting to produce lethal offspring or lead to the extinction of certain dominated niches. Two population members P_i and P_j, with corresponding centers $\mathbf{c}_i$ and $\mathbf{c}_j$, are considered to be coming from different niches if neither one of them is within the other member's niche. That is, P_j is not within P_i's niche, or $dist(P_i, P_j) > K\sigma_i^2$; and P_i is not within P_j's niche, or $dist(P_i, P_j) > K\sigma_j^2$, where $dist\,(P_i, P_j) = \|\mathbf{c}_i - \mathbf{c}_j\|^2$. These two conditions are equivalent to $dist(P_i, P_j) > K \max\left(\sigma_i^2, \sigma_j^2\right)$. If parents from different niches are forbidden from mating regardless of their fitness values, then mediocre individuals located in non-niche areas of the search space and which generally result from the random initialization process will not be allowed to improve. Therefore, mating restriction should be relaxed by allowing unconditional mating between two individuals if at least one of them has low fitness. The mating restriction rule for individuals P_i and P_j is summarized as follows

IF $\left(dist\,(P_i, P_j) > K \max\left(\sigma_i^2, \sigma_j^2\right)\right.$ and $f_i > f_{min}$ and $\left.f_j > f_{min}\right)$ THEN no crossover.

When mating is allowed, the crossover of two individuals is performed independently on each of the string sections consisting of the individual feature dimensions of the candidate cluster center. This leads to n independent crossovers per offspring. After crossover, each bit in an offspring individual's chromosome string can be inverted with a small mutation probability P_m.

D. Scale Estimation

The scale parameter or niche radius that maximizes the fitness value for the i^{th} cluster can be found by setting $\frac{\partial f_i}{\partial \sigma_i^2} = 0$ to obtain

$$\sigma_i^2 = \frac{\sum_{j=1}^{N} w_{ij} d_{ij}^2}{\sum_{j=1}^{N} w_{ij}}. \tag{2}$$

Therefore, σ_i^2 will be updated using an iterative hill-climbing procedure, using the previous values of σ_i^2 to compute the weights w_{ij} in (2). Note that this approximation is reasonable because by virtue of the replacement scheme in DC, the centers for an individual are not expected to change drastically from one generation to the next. Even though the exponential weights w_{ij} decrease very fast towards zero with distance, it is important to note that each cluster center still sees all the points in the data set through these soft (non-binary) weights. We map the weights to binary values, using a suitable threshold value, T_w, to allow only the core members of the cluster to be involved in the estimation of its parameters. The binarization is done as follows

$$w_{ij} = \begin{cases} 1 & \text{if } w_{ij} > T_w \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

When mating takes place, each child should inherit the scale parameter, σ_i^2, of the closest parent as its initial scale. This inheritence is possible because in DC, children are most likely to replace similar parents. This makes it possible to integrate a hill-climbing step that encompasses all generations for the scale parameter updates. This means that the scale parameters from one generation are used as a starting point to update the scale parameters in the next generation. Following their inheritence by both children, the scale parameters are updated using (2). Similarly, the parents' scale parameters are updated so that both children and parents' scale parameters would have implicitly undergone the same number of updates starting from the initial generation. This is essential to make a fair comparison of their fitnesses for the replacement decision in DC. After the scale updates, the fitness values are computed for parents and children. When mating is not allowed, the parents' scale parameters are updated, their fitness values are recomputed, and they remain in the population for the next generation.

E. Degenerate and Spurious Solutions

When an individual in the population is initialized at a remote location in a data set, it will try to expand by increasing its scale parameter as much as possible so that it will include more inliers and hence increase its fitness. We can limit this behavior by ensuring that its scale parameter does not exceed a theoretical upper bound on σ_i^2 that can be shown to be approximately

$$\sigma_{max}^2 = \frac{\sum_{p=1}^{n} \left(\max_{j=1}^{N} x_{j_p} - \min_{j=1}^{N} x_{j_p}\right)^2}{4 \times \chi_{2,.995}^2}. \quad (4)$$

The population may also contain spurious clusters with very few points which form their own niches because they tend to be located in the isolated and sparse areas of the feature space. Therefore we modify the fitness function to become

$$f_i = \begin{cases} \frac{\sum_{j=1}^{N} w_{ij}}{\sigma_i^2} & \text{if } \sigma_i^2 \leq \sigma_{max}^2 \text{ and} \\ & \sum_{j=1}^{N} w_{ij} \geq N_{min} \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

where N_{min} should be equal to the lowest acceptable cluster cardinality. Consequently, an individual with zero fitness can never survive into the next generation because in DC, a better fit child will always replace its parent. In addition, we prevent the child that is most similar to a zero fitness parent from replacing this parent to limit the propagation of mediocrity into future generations. This is done by modifying the DC replacement rule whenever a zero fitness or invalid parent is selected to contribute in a crossover operation. In this case, regardless of whether crossover takes place, the invalid parent is replaced by the other parent. However, if the latter is also invalid, then it is replaced by a valid member that is randomly selected from the population. Note that we allow zero fitness parents to reproduce when mating is not restricted because by mating with individuals from different niches, it is possible to generate individuals in new unexplored areas of the feature space, particularly in the early generations.

F. Extracting the Final Cluster Centers

After convergence of the population, almost all individuals converge to the niche peaks or cluster centers. At this point, we extract the best individual from each good niche to obtain the set of final cluster centers, $\mathcal{C}$. The extraction is done using a greedy approach, while ensuring that no two extracted centers are similar. Two candidate centers are considered to be similar if each one of them is within the other one's niche. The extraction procedure is presented below:

Final Cluster Center Extraction

Sort individuals P_i *to obtain* $P_{(i)}$, $i = 1, \cdots, N_P$, *such that* $f_{(1)} \geq f_{(2)} \geq \cdots \geq f_{(N_P)}$;
Initialize set of cluster centers $\mathcal{C} = \{P_{(1)}\}$;
FOR $i = 2$ **TO** N_P **DO** {
  IF $f_{(i)} > f_{min\_extract}$ *AND*
   $dist\left(P_{(i)}, P_{(k)}\right) > K \min\left(\sigma_i^2, \sigma_k^2\right) \quad \forall P_{(k)} \in \mathcal{C}$
   THEN $\mathcal{C} \leftarrow \mathcal{C} \cup P_{(i)}$;
}

We refer to this new approach to unsupervised clustering as the Unsupervised Niche Clustering algorithm (UNC).

G. Computational Complexity

In each generation, the most extensive computational requirement for UNC consists of computing the residuals, fitness and scale, for each of the N_P individuals in the population, and the inter-niche distances, resulting in $\mathcal{O}\left(N_P\left(N + N_P\right)\right)$ computations. The extraction step requires sorting the fitnesses and computing the inter-niche distances, and hence is $\mathcal{O}\left(N_P\left(\log N_P + N_P\right)\right)$.

H. Refinement of the Extracted Prototypes

While GAs are superior in terms of their ability to do a global search for the optima, their solutions may suffer from inaccuracy. For this reason, it is recommended that a local search be performed in the neighborhood of each solution that is provided by Genetic Optimization (GO). To make the local refinement of the parameters of each cluster independent of other clusters, the data set is partitioned into C clusters, $\mathcal{X}_i$, $i = 1, \cdots, C$, before performing the local search, such that each feature vector is assigned to the closest prototype. Subsequently, a gradient ascent search can be used to refine the center and scale estimates found by GO. The optimal scale estimates can be derived as in Section III-D to obtain

$$\sigma_i^2 = \frac{\sum_{\mathbf{x}_{(j)} \in \mathcal{X}_i} w_{ij} d_{ij}^2}{\sum_{\mathbf{x}_{(j)} \in \mathcal{X}_i} w_{ij}}. \quad (6)$$

Similarly, the centers can be optimized by setting $\frac{\partial f_i}{\partial \mathbf{c}_i} = 0$ to obtain

$$\mathbf{c}_i = \frac{\sum_{\mathbf{x}_{(j)} \in \mathcal{X}_i} w_{ij} \mathbf{x}_j}{\sum_{\mathbf{x}_{(j)} \in \mathcal{X}_i} w_{ij}}. \quad (7)$$

The refinement of the GO solution will consist of alternating optimization updates of σ_i^2 and $\mathbf{c}_i$ using

(6) and (7) for a few iterations, with a recomputaion of the weights w_{ij} before each update. We refer to this refinement scheme as Density Fitness Gradient Ascent (DFGA).

The above alternating optimization scheme results in accurate center estimates. However, the scale parameters tend to be overestimated, particularly in the presence of noise and when clusters are very close to each other. In [10], we presented a new robust estimator, called the Maximal Density Estimator (MDE) which can estimate the center and scale parameters accurately and efficiently using an alternating optimization of the centers as given by (7) and scale parameters as follows

$$\sigma_i^2 = \frac{\sum_{\mathbf{x}_{(j)} \in \mathcal{X}_i} w_{ij} d_{ij}^4}{3 \sum_{\mathbf{x}_{(j)} \in \mathcal{X}_i} w_{ij} d_{ij}^2}. \qquad (8)$$

We have noticed that final refinement using MDE yields center and scale estimates that are significantly more accurate than the previous scheme.

IV. Experimental Results

We present results on two data sets shown on Figs. 2(a) and 3(a). The total number of points, N, and the number of noise points, N_n, for these Gaussian clusters are listed in the first column of Table I. The generating center coordinates were: (50, 50), (90, 90), and (150, 150) for Data set No. 1 and (50, 50), (80, 80), (80, 150), (150, 100), (160, 170), and (200, 200) for Data set No. 2.The GA parameters were: population size = 200, no. of generations = 200, $P_c = .9$, $P_m = 5 \times 10^{-6}$, $K = \chi^2_{2,.995}$, $f_{min} = 0.6$, $f_{min\_extract} = 0.3$, and $T_w = 0.3$. Fig. 1 shows the evolution of the population (denoted by square symbols) using UNC for Data Set No. 2. The initial population is chosen randomly from the set of feature vectors. This explains the higher concentration of solutions in the densest areas, which converge toward the correct centers in subsequent generations. The cluster centers found using UNC for the two data sets, along with the refined estimates by DFGA and MDE, are listed in the last three columns of Table I, and shown in Figs. 2 and 3. In these figures, the circular contours around each cluster center depict the normalized distances, $\frac{d_{ij}^2}{\sigma_i^2}$, corresponding to $\chi^2_{0.25}$, $\chi^2_{0.5}$, $\chi^2_{0.75}$, and $\chi^2_{0.975}$. Therefore, they reflect the accuracy of the final scale estimates.

V. Conclusion

In this paper, we presented a new unsupervised robust clustering algorithm based on genetic niching. Using an appropriate density fitness measure,

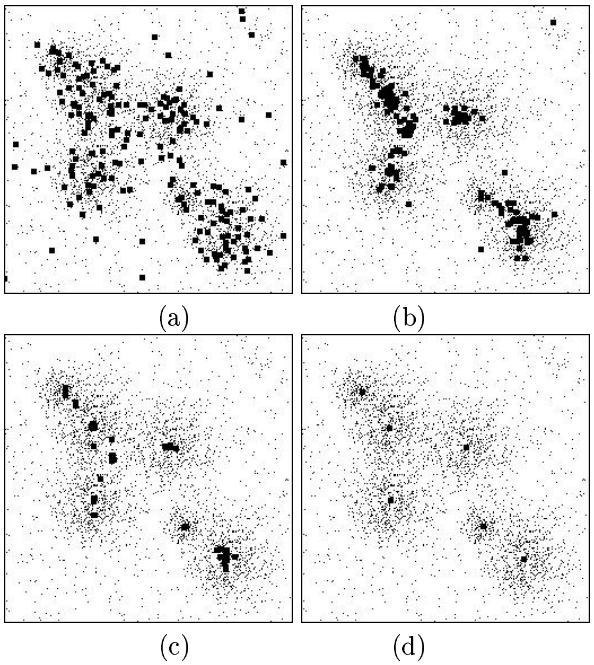

Fig. 1. Evolution of the population using UNC (a) Initial population, (b) population after 20 generations, (c) population after 200 generations, (d) extracted centers

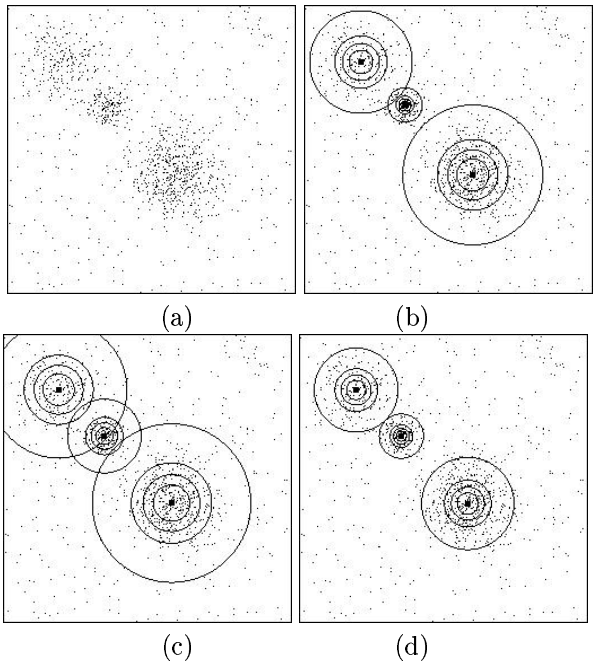

Fig. 2. Clustering Results for Data set No. 1 with UNC (a) Data set No. 1, (b) extracted prototypes, (c) prototypes refined using DFGA, (d) prototypes refined using MDE

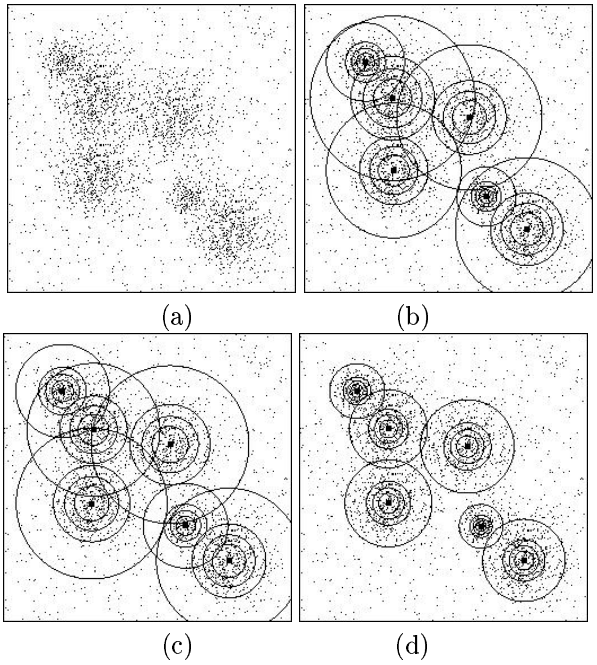

Fig. 3. Clustering Results for Data set No. 2 with UNC (a) Data set No. 2, (b) extracted prototypes, (c) prototypes refined using DFGA, (d) prototypes refined using MDE

we transform clusters in feature space into niches in the fitness landscape. The niche peaks, which represent the final cluster centers, are identified based on Deterministic Crowding. We eliminate the problem of crossover interactions in DC by allowing mating only between members within the niche. For this purpose, we estimate the niche radii by using an iterative hill-climbing procedure coupled with the genetic optimization of the cluster centers. We refine the extracted centers using local optimization methods such as DFGA or MDE. The examples illustrate the effectiveness of the approach in determining clusters in noisy data. Because our approach does not necessitate the derivation of an analytical solution for the cluster prototypes, it offers the advantage of being able to handle more general distance measures. In particular, non metric dissimilarity measures can easily be incorporated for general applications such as Web or data mining and information retrieval. The current implementation for our new approach uses Euclidean distance, and is therefore optimal for spherically distributed clusters. However, it can easily be extended to more general distributions. For instance, when the clusters are expected to be ellipsoidally shaped, a covariance matrix can be used instead of a single scale parameter, in conjunction with a Mahalanobis type distance in the computation of fitness values.

TABLE I

DATA SET PROPERTIES AND RESULTS FOR UNC EXPERIMENTS

No.	$N(N_n)$	UNC	UNC + DFGA	UNC + MDE
1	1157(346)	(50, 50)	(48.8, 48.9)	(50.1, 49.1)
		(89, 88)	(89.3, 89.9)	(90.1, 90.0)
		(149, 150)	(148.8, 149.3)	(149.0, 149.7)
2	2530(626)	(54, 51)	(52.2, 51.0)	(51.0, 50.9)
		(78, 83)	(79.7, 85.2)	(78.7, 84.4)
		(79, 147)	(78.1, 150.6)	(78.8, 150.3)
		(146, 100)	(147.5, 98.4)	(148.6, 100.1)
		(161, 170)	(161.2, 170.3)	(160.7, 170.7)
		(197, 199)	(199.7, 201.4)	(198.7, 201.0)

Acknowledgment

Partial support of this work by the National Science Foundation Grant IIS 9800899 is gratefully acknowledged.

REFERENCES

[1] J. C. Bezdek, S. Boggavarapu, L. O. Hall, and A. Bensaid, "Genetic algorithm guided clustering," in *First IEEE conference on evolutionary computation*, Orlando, Florida, June. 1994, vol. 1, pp. 34–39.

[2] O. Nasraoui and R. Krishnapuram, "Clustering using a genetic fuzzy least median of squares algorithm," in *North American Fuzzy Information Processing Society Conference*, Syracuse NY, Sep. 1997.

[3] J. M. Jolion, P. Meer, and S. Bataouche, "Robust clustering with applications in computer vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 8, pp. 791–802, Aug. 1991.

[4] R. Krishnapuram and C.-P. Freg, "Fitting an unknown number of lines and planes to image data through compatible cluster merging," *Pattern Recognition*, vol. 25, no. 4, 1992.

[5] H. Frigui and R. Krishnapuram, "Clustering by competitive agglomeration," *Pattern Recognition*, vol. 30, no. 7, pp. 1109–1119, 1997.

[6] J. H. Holland, *Adaptation in natural and artificial systems*, MIT Press, 1975.

[7] D. E. Goldberg and J. J. Richardson, "Genetic algorithms with sharing for multimodal function optimization," in *2nd Intl. Conf. Genetic Algsorithms*, Cambridge, MA, Jul. 1987, pp. 41–49.

[8] K. A. De Jong, "An analysis of the behavior of a class of genetic adaptive systems," *Doct. Diss., U. of Michigan.*, vol. 36, no. 10-5140B, pp. 29–60, 1975.

[9] S. W. Mahfoud, "Crowding and preselection revisited," in *2nd Conf. Parallel problem Solving from Nature, PPSN '92*, Brussels, Belgium, Sep. 1992.

[10] O. Nasraoui and R. Krishnapuram, "A robust estimator based on density and scale optimization, and its application to clustering," in *IEEE International Conference on Fuzzy Systems*, New Orleans, LA, Sep. 1996, pp. 1031–1035.