On Fitness, Niching Strategies, and Hybrid Niche Size Estimation for Discovering an Unknown Number of Clusters in Noisy Data

Olfa Nasraoui and Elizabeth Leon

Department of Electrical and Computer Engineering, The University of Memphis
Memphis, TN 38152
{onasraou, eleon}@memphis.edu

Abstract. Clustering is crucial to many applications in pattern recognition, data mining, and machine learning. Evolutionary techniques have been used with success in clustering. However, most existing evolutionary clustering techniques still suffer from several drawbacks. After surveying existing evolutionary clustering techniques, we show that **(i)** robustness to noise can be achieved with a robust fitness measure, while **(ii)** scalability of the search space with respect to the number of clusters and to the size of the data can be achieved by encoding a single cluster prototype in the chromosome, **(iii)** the resulting multimodal optimization problem should be solved using a *niching* strategy, which will also allow the determination of the number of clusters automatically, and finally that **(iv)** a hybrid Piccard niche size estimation strategy is the key to implement a sound mating restricion, and is the key to successful fitness sharing.

1 Introduction and Motivations

Clustering [1] is an effective technique for data mining and exploratory data analysis that aims at classifying the unlabeled points in a data set into different groups or clusters, such that members of the same cluster are as similar as possible, while members of different clusters are as dissimilar as possible. The Simple Genetic Algorithm (SGA) has been successfully used to search the solution space in clustering problems with a *fixed* number of clusters [2], and for robust clustering [3]. However, in practice, the number of clusters is rarely known, and the data sets can be contaminated with noise. In this paper, we survey some existing *evolutionary* clustering techniques, showing that they suffer from one or more of the following shortcomings: **(i)** they are *not robust* in the presence of noise, **(ii)** the size of the search space *explodes exponentially* with the number of clusters, or with the number of data points, and **(iii)** they assume a *known* number of clusters. We then discuss the requirements to achieve the above objectives, and illustrate our arguments with empirical simulations.

This paper is organized as follows. In Section 2, we survey some evolutionary clustering techniques. In Section 3, we discuss some basic requirements for evolutionary clustering. In Section 4, we describe a complete approach that satisfies the choices of optimal chromosome encoding and robust fitness measure discussed in the previous sections. In Section 5, we present our experimental results. In Section 6 we study the effect of the selected niching strategy and propose a Hybrid Niche Size Estimation, based on Piccard updates of the niche sizes, that dramatically improves Fitness Sharing for clustering. In Section 7, we study the role of the fitness measure, and compare the density-based fitness to the Sum of Squared Errors fitness. Finally, we present our conclusions in Section 8.

2 An Overview of Existing Evolutionary Clustering Techniques

One of the earliest attempts at using a GA for clustering was made by Raghavan and Birchand [4]. In their approach, the GA was used to optimize the square error of clustering (similar to K Means's criterion). Each chromosome represented a possible partition of the entire data set consisting of N objects. Hence the chromosome consisted of N substrings, with each substring encoding one of c cluster labels. This encoding led to an explosion of the search space size as the data set got larger, and assumed a known number of clusters. Also, the n-point crossover frequently resulted in meaningless or lethal partitions; and the square error based fitness function meant that the approach was sensitive to noise. Bhuyan et al. [5] proposed an improved encoding that used a separator symbol ($*$) to separate the clusters, consisting of a string of data object labels, and Goldberg's permutation crossover [6] to yield valid offspring. However, the solution suffered from en explosion in permutation redundancy because of the arbitrary order of data labels. Babu and Murty [7] used the GA only to find good initial solutions, and then used K-Means for the final partition. This was the first hybrid clustering approach that obviously outperformed the use of either K-means or GA alone. However, it is not resistant to noise, and assumed a known number of clusters. Fogel and Simpson [8] use Evolutionary Programming to solve the fuzzy min-max cluster problem by directly encoding the centroids (instead of the entire partition) in the chromosome string. This was undoubtedly one of the first efficient encodings for the clustering problem, and it influenced most subsequent evolutionary clustering methods. Hall et al. [9, 2] proposed a genetically guided approach (GGA) to optimizing the reformulated Hard and Fuzzy C-Means (HCM and FCM) objective functions. The chromosome strings encode the c center vectors (of p individual features per vector) of the candidate solutions, and a standard GA evolves the population. However, GGA is *not robust* in the face of noise, and it *can not determine the number of clusters automatically.* Also, because all c cluster centers are encoded in each chromosome string, the size of the search space *explodes exponentially with the number of clusters.* In [3], a robust estimator was proposed, that is based on the LMedS that can simultaneously partition a given data set into c clusters, and estimate their parameters. In addition to its limitation of estimating the parameters of a single structure, the LMedS suffers from a major drawback in that it has a nonlinear, and nondifferentiable objective function that is not amenable to mathematical or numerical optimization. For this reason, we proposed the integration of a genetic algorithm to the partitioning and estimation process, in order to search the solution space more efficiently. This resulted in a new approach to *robust genetic clustering* based on LMedS. However, the technique still assumed that the number of clusters was *known in advance*, and that the *noise contamination rate was* 50%. Also, because all c cluster centers are encoded in each chromosome string, the size of the search space explodes exponentionally with the number of clusters as in the case of GGA. Lee and Antonsson present an algorithm for unsupervised clustering using Evolutionary Strategies (ES) in [10]. In their approach, all cluster centroids are coded into a variable length chromosome, and only crossover is used to vary the number of clusters. However this coding scheme suffers from an exponential increase in the complexity of search with the number of clusters, and is also not robust to noise and outliers because its fitness measure is based on a classical sum of errors. Rousseeuw's original robust K-Medoid criterion [11] was optimized in [12] using a hybrid GA approach. Though more robust to noise, this approach assumes a known number of clusters. Also, because the cluster representatives are medians, it is most efficient when the rate of noise is exactly 50%, and cannot adapt to various noise contamination rates. It also does not have any provision for clusters of different sizes since it has no notion of scale, and the size of the search space explodes exponentionally with the number of clusters. The Unsupervised Niche Clustering (UNC) [13] is a recent approach to unsupervised robust clustering based on Genetic optimization. UNC uses a chromosome representation encoding a single cluster prototype , and optimizes a density based fitness function that reaches a maximum at every good cluster center. Hence, UNC uses Deterministic Crowding (DC) [14] for niching. To further alleviate the problem of crossover interaction between distinct niches, UNC relies on an improved restricted mating scheme which relies on an accurate and assumption-free estimate of the niche radii which are not restricted to be equal for all peaks. Because UNC uses robust weights in its cluster fitness definition, it is less sensitive to the presence of noise. Furthermore, the combination of the single-cluster chromosome encoding with niching offers a simple and efficient approach to automatically determine the optimal number of clusters.

Table 1 compares some evolutionary clustering techniques, showing that UNC has all the desired features lacking in other approaches. Note that a *partitional* approach relies on a sum of error type objective function that requires encoding of c cluster centers or a long c-ary partition string of length N. Partitional approaches also require the additional costly overhead of *re-partitioning the data points into c clusters with each fitness computation.* On the other hand, a *density* based approach directly optimizes each cluster density independently of other clusters, hence eliminating the need to partition data. Density is also a more sensible measure of cluster validity, and is naturally resistant to noise. Complexity is listed per generation. Also hybrid approaches tend to converge in fewer generations compared to purely evolutionary search methods.

Table 1. Comparison of UNC with Other Evolutionary Clustering Algorithms for data of size N, population of size N_P, and C clusters

Approach →	*UNC* [13]	*GGA* [2]	*Lee* [10]	*G-C-LMedS* [3]	*k-d-Median* [12]
Search Method	GA	GA	ES	GA	GA
Robustness to noise	yes	no	no	yes	yes
Automatic Scale Estimation	yes	no	no	no	no
Complexity	$O(NN_P)$	$O(CNN_P)$	$O(CNN_P)$	$O(CNN_P)$	$O(N_PCN\log(N))$
Hybrid	yes	no	no	no	yes
Does not require No. of Clusters	yes	no	yes	no	no
Handles ellipsoidal clusters	yes	no	no	no	no

3 Some Basic Requirements for Evolutionary Clustering

3.1 The Need for a Robust Fitness Measure

Most existing evolutionary clustering techniques, such as [4], [7], [8], [9, 2], and [10], rely on a fitness that is based on a Sum of Squared Errors like most prototype based clustering methods, such as the $K-$Means and its fuzzy counterpart, the Fuzzy $C-$Means (FCM) [15] algorithms. Let $\mathcal{X} = \{\mathbf{x}_j | j = 1, \ldots, N\}$ be a set of feature vectors in an $n-$dimensional feature space with coordinate axis labels $[x_1,\ x_2,\ \ldots,\ x_n]$, where $\mathbf{x}_j = [x_{j1},\ x_{j2},\ \ldots,\ x_{jn}]$. Let $\mathbf{B} = (\beta_1, \ldots, \beta_c)$ represent a C-tuple of prototypes each of which characterizes one of the C clusters. Each β_i consists of a set of parameters. The $K-$Means has the following objective $J = \sum_{i=1}^{C} \sum_{\mathbf{x}_j \in \mathcal{X}_i} d_{ij}^2$, where $d_{ij}^2 = d^2(\mathbf{x}_j, \beta_i)$ represents the distance from a feature point $\mathbf{x}_j$ to the prototype β_i, and $\mathcal{X}_i$, the i^{th} cluster, is given by $\mathcal{X}_i = \left\{\mathbf{x}_j \in \mathcal{X} |\ d_{ij}^2 = \min_{k=1}^{C} d_{kj}^2\right\}$. It can be seen that K-Means minimizes the sum of squared errors for each cluster (i), and that its solution is essentially based on a Least Squares (LS) approach, yielding as a solution the mean vector or centroid of each cluster. Unfortunately, the ordinary Least Squares (LS) method to estimate parameters is not robust because its objective function, $\sum_{j=1}^{N} d_j^2$, increases indefinitely with the distance d_j. Hence, extreme outliers with arbitrarily large residuals can have an infinitely large influence on the resulting estimate. For this reason, we conjecture that in order to decrease the influence of outliers, the cluster fitness measure must be based on a *robust* estimator, such as an M-estimator [16].

3.2 The Need for a Scalable Chromosome Encoding

Most existing evolutionary clustering techniques are based on one of the following two chromosome encodings. In the first encoding, each chromosome represents a possible partition of the entire data set consisting of N objects. Hence the chromosome consists of N substrings, with each substring encoding one of C cluster labels. Obviously this encoding leads to an explosion of the search space size as the data set gets larger. Moreover, the use of a limited alphabet size to encode the cluster label assumes a known number of clusters. In the second approach, the chromosome string encodes the C center prototypes (such as p individual features per vector) of the candidate solutions. In addition to assuming a known number of clusters, the size of the search space *explodes exponentially with the number of clusters.* The only way to achieve an encoding that is scalable (i.e., a search space size that grows linearly) with respect to both the number of clusters and the size of the data, is to make it independent of these two charateristics, and the only way to realize this objective is to encode a single cluster prototype in each chromosome regardless of the size of the data or of the number of clusters.

3.3 On the Need for Niching and Automatic Niche Size Estimation

The optimal single cluster chromosome encoding strategy will cause the landscape of any fitness function to reach several suboptimal peaks (multiple modes) located at the centroids of these clusters, which makes their identification a multi-modal optimization problem. Therefore, niching methods are required to identify the multiple optima. As in nature, niches in our context correspond to different subspaces of the environment (clusters) that can support different types of life (data samples). Sharing methods [17, 18] attempt to maintain a diverse population by reducing the fitness of individuals that have highly similar members within the population. This in turn discourages redundant solutions from overtaking the entire population, while rewarding individuals that uniquely exploit specific areas of the domain. Crowding methods [19], try to form and maintain niches by replacing population members preferrably with the most similar individuals. Mahfoud [14] proposed an improved crowding mechanism, called *"Deterministic Crowding" (DC)*, which is free of any parameter. However, without a proper mating restriction, both sharing and DC risk losing niches due to *crossover interaction* after many generations [14].

4 Unsupervised Robust Clustering using Genetic Niching

The clustering problem was reformulated in [13], by modifying the objective from searching the solution space for C clusters to searching this space for any one cluster. The fitness value, f_i, for the i^{th} candidate center location, $\mathbf{c}_i$, is defined as the density of a hypothetical cluster at that location, defined as $f_i = \frac{\sum_{j=1}^{N} w_{ij}}{\sigma_i^2}$, where $w_{ij} = \exp -\frac{d_{ij}^2}{2\sigma_i^2}$ is a robust weight that measures how typical data point $\mathbf{x}_j$ is in the i^{th} cluster, σ_i^2 is a robust measure of scale (dispersion) for the i^{th} cluster, d_{ij}^2 is the distance from data point $\mathbf{x}_j$ to cluster center $\mathbf{c}_i$, and N is the number of data points. The landscape of the density fitness function is expected to reach several suboptimal peaks (multiple modes) located at the centroids of these clusters, and their identification is a multi-modal optimization problem. Therefore, niching methods, which can identify multiple optima within multimodal domains, are a necessity. For the clustering problem, "Deterministic Crowding" (DC) [14] was found to work best. It can easily be seen that as a variance measure, σ_i^2 is also related to the radius of the niche, since in this problem, each cluster in the data set will generate a niche in the fitness landscape. Note that the robust weights w_{ij} will be small for outliers, hence offering a means of distinguishing between good data and noise.

4.1 Crossover and Mutation

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

4.2 Hybrid Piccard Niche Size Estimation and Mating Restriction

The scale parameter that maximizes the fitness value for the i^{th} cluster can be found by setting $\frac{\partial f_i}{\partial \sigma_i^2} = 0$, and assuming that the weights, w_{ij}, do not change drastically from one generation to the next, to obtain the piccard update equations: $\sigma_i^2 = \frac{\sum_{j=1}^{N} w_{ij} d_{ij}^2}{\sum_{j=1}^{N} w_{ij}}$. Therefore, σ_i^2 will be updated once per generation, using the previous (i.e. inherited from the closest parent) values of σ_i^2 to compute the weights w_{ij}, and the initial values $\sigma_i^2 = \frac{\sigma_{max}^2}{K_\sigma}$; where σ_{max}^2 is an estimate of the maximal radius of the data set, and $K_\sigma = 10$ This *hybrid* genetic optimization converges much faster than a purely genetic search. The above niche size estimates delineate the contours of the niches. Hence, they can be used to implement a simple mating restriction scheme based on preventing crossover between individuals from different niches. Individuals P_i and P_j are considered from different niches if their phenotypical Euclidean distance exceeds $Kmax(\sigma_i^2, \sigma_j^2)$.

4.3 Extracting Cluster Centers From the Final Population

After convergence of the population, we extract the best individual from each good niche to obtain the set of final cluster centers, $\mathcal{C}$, as shown below:

```
Final Cluster Center Extraction
Sort individuals $P_i$ in descending order of their fitness values 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 $\mathcal{C} = \emptyset$;
FOR $i = 1$ TO $N_P$ DO {
   IF $f_{(i)} > f_{min\_extract}$ AND $P_i$ and $P_k$ are from different niches $\forall P_{(k)} \in \mathcal{C}$ THEN
      $\mathcal{C} \leftarrow \mathcal{C} \cup P_{(i)}$;
```

4.4 Final Local Piccard Refinement of the Extracted Prototypes

It is recommended that a local search be performed in the neighborhood of each solution provided by Genetic Optimization (GO) to increase accuracy, without the excessive cost of a large population or of too many generations. To make the local refinement of the parameters of each cluster independent of other clusters, the data set is partitioned into c clusters before performing the local search, such that each feature vector is assigned to the closest prototype. Subsequently, the i^{th} cluster is given by $\mathcal{X}_i = \left\{ \mathbf{x}_{(k)} \in \mathcal{X} \mid d_{ik}^2 < d_{jk}^2 \ \forall j \neq i \right\}$, for $1 \leq i \leq c$. In [20], we presented a new iterative robust estimator, called the Maximal Density Estimator (MDE) which can estimate the center and scale parameters accurately and efficiently (with linear complexity). MDE uses fast alternating Piccard updates of the centers using $\mathbf{c}_i = \frac{\sum_{\mathbf{x}_{(j)} \in \mathcal{X}_i} w_{ij} \mathbf{x}_j}{\sum_{\mathbf{x}_{(j)} \in \mathcal{X}_i} w_{ij}}$, and the scale parameters using $\sigma_i^2 = \frac{\sum_{\mathbf{x}_{(j)} \in \mathcal{X}_i} w_{ij} d_{ij}^4}{3 \sum_{\mathbf{x}_{(j)} \in \mathcal{X}_i} w_{ij} d_{ij}^2}$.

4.5 Combining Fitness Sharing with Hybrid Piccard Niche Size Estimation

Sharing methods [18] rely strongly on correct estimates of the niche counts, i. e., the number of individuals in each niche. The niche counts themselves depend on a parameter, σ_{sh}, which ideally, should approximate the widths of the peaks. We propose to use the Piccard scale update strategy in Section 4.2 to automatically estimate individualized σ_{sh}, for each individual.

5 Simulation Results

5.1 Detailed Phases of Cluster Evolution

Fig. 1 shows the evolution of the population (denoted by square symbols) using UNC for a noisy data set with 5 clusters. 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 leveled contours correspond to boundaries including increasing quantiles ($\alpha = 0.25, 0.5, 0.75, 0.99$) as derived from the normalized distance values $\frac{d^2}{\sigma_i^2} = \chi^2_{2,\alpha}$. Hence, they indicate the accuracy and robustness of UNC's cluster scale estimates σ_i^2.

5.2 Sensitivity to GA Parameters, Noise, and Effect of Final Refinement

Our simulation results using various population sizes showed similar results as long as the population size exceeded 50 individuals. Convergence of the population occured relatively fast, after about 30 generations. Hence, we show the

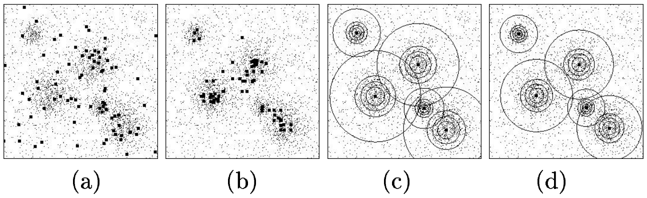

(a) (b) (c) (d)

Fig. 1. Evolution of the population using UNC: (a) Initial population, (b) population after 30 generations, (c) extracted centers and scales, (d) final centers and scales after refinement

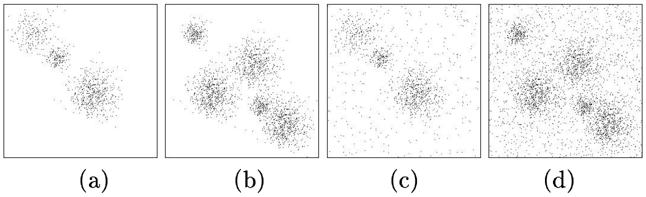
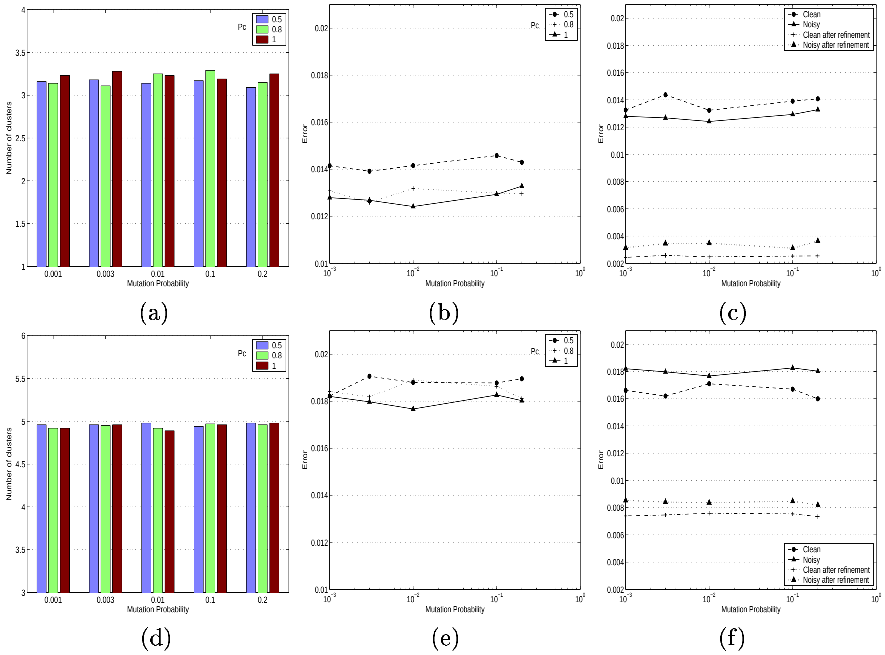

(a) (b) (c) (d)

Fig. 2. Data Sets with 3 and 5 clusters: (a,b) Clean data sets (c,d) Noisy data sets



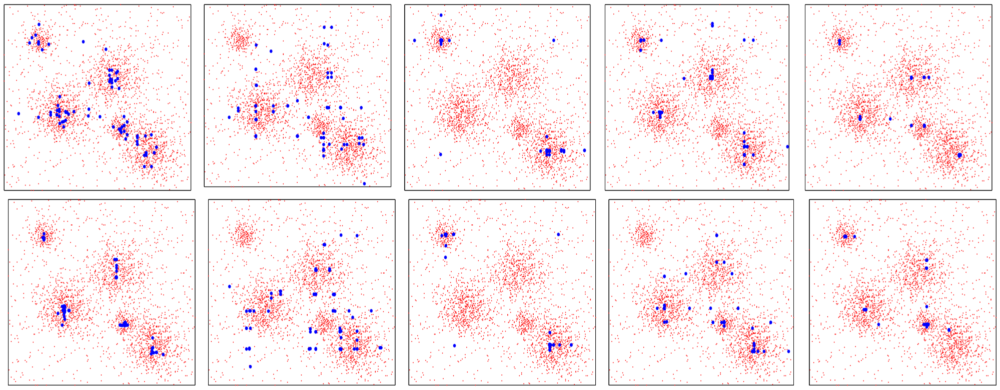
(a) (b) (c)
(d) (e) (f)

Fig. 3. Results of UNC for 3 clusters with noise (1^{st} row) and 5 clusters with noise (2^{nd} row), averaged over 100 runs: (a,d) Average number of cluster for different crossover and mutation probabilities; (b,e) Center Error; (c,f) Center Error before and after refinement

effect of the only GA parameters that seemed to affect the results: crossover probability (P_c), and mutation probability (P_m). We performed 100 runs and show the averages of the following clustering quality measures: **(i)** Number of final extracted clusters, and **(ii)** Average normalized centroid error. The GA parameters throughout all the experiments in this paper were: population size = 80, number of generations = 30. We start by studying the effect of crossover and mutation as well as refinement on the clean and noisy data sets depicted in Fig. 2. Figs. 3(a) and (d) show that for a wide range of crossover and mutation rates, UNC can discover the correct number of clusters in the vast majority of the experiments. Figs. 3(b) and (e) show that the accuracy of the center estimates, *even without any final local refinement*, is within 1.5% of the range of the images, and that this accuracy improves with an increased crossover rate, while being stable for a wide range of mutation rates. The former is due to the fact that crossover encourages local exploration and recombination of good solutions to improve converged solutions. While the latter is due to Deterministic Crowding's replacement strategy which prevents lethal children resulting from mutation, from replacing their parents. The top two curves in each of Figs. 3(c) and (f) show that UNC is quite robust to noise, yielding comparable unrefined center estimate accuracies for both clean and noisy versions of the data sets, while the bottom two curves confirm that final local refinement dramatically improves this accuracy, as expected.

6 Effect of the Selected Niching Strategy and Hybrid Niche Size Estimation: The Case of Fitness Sharing

To illustrate the difficulty in using fitness sharing [17, 18] when chosen as the niching mechanism, we replaced the DC replacement with fitness sharing combined with mating restriction based on line breeding. Fig. 4 compares the evolution of the population of cluster centroids (bold circles) using DC (column (a)) versus using fitness sharing, with two σ_{sh} values in the (columns (b) and (c), respectively). Note that being of unequal sizes and densities, the clusters will generate peaks of different widths and heights, ruling out any existing method to pre-estimate the value of σ_{sh}. In fact, for such a problem, it is clear that there is no *single* value of σ_{sh} that would be optimal, but rather a different value is needed for each niche. Nevertheless, we assume that this parameter is known, and show experiments with a Proximity factor $PF = 1$ and $\sigma_{sh} = 0.01$ corresponding to the niche size for the smallest cluster, and $\sigma_{sh} = 0.4$ roughly corresponding to the niche size for the largest cluster. DC, with the automatic Piccard scale updating and restricted mating, shows a very good within peak convergence for each cluster in Fig. 4 (a). Fig. 4 (b) shows that fitness sharing with $\sigma_{sh} = 0.01$ evolves a good solution only for the smallest clusters, but cannot keep it beyond 30 generations. Each of the remaining big clusters is broken into several small peaks. Fig. 4 (c) shows that fitness sharing with $\sigma_{sh} = 0.4$ shows an improvement of within peak convergence, but only two out of the five peaks survive beyond 30 generations. Finally, we show the results of combining fitness sharing with our Piccard scale updating strategy for estimating distinct σ_{sh} values for each individual in the population, in Fig. 4 (d) using line breeding; and in Fig. 4 (e) using UNC's mating restriction. It is clear that the results are better than those of standard sharing (with a fixed σ_{sh}) in terms of within peak convergence and survival of a diverse set of solutions. Furthermore, we notice that sharing combined with UNC's mating restriction (Fig. 4 (e)) is able to maintain all the niches, while sharing with line breeding (Fig. 4 (d)) is unable to maintain all niches beyond 30 generations.



(a) (b) (c) (d) (e)

Fig. 4. Evolution of the population after 30 generations (1^{st} row) and after 800 generations (2^{nd} row), using (a) Deterministic Crowding versus (b) Sharing with $\sigma_{sh} = 0.01$, (c) Sharing with $\sigma_{sh} = 0.4$, (d) Sharing with UNC's Hybrid Piccard scale updating strategy for estimating distinct σ_{sh} values for each individual and line breeding, and (e) Sharing with UNC's Hybrid Piccard scale updating strategy for estimating distinct σ_{sh} values for each individual and UNC's mating restriction

7 Importance of a Robust Fitness Measure

To illustrate the inadequacies of a non-density based fitness such as one that is based on a sum of squared errors, we replaced our density based fitness measure in UNC by the Sum of Squared Errors in the candidate cluster. This is the kind of fitness that guides the evolutionary search in most existing evolutionary clustering approaches [7–9, 2, 10], as well as the classical K-Means clustering algorithm. The results in Fig. 5 (2nd row) show that this non-robust fitness, regardless of the niching strategy, is simply inadequate for noisy data sets, since its global optimum is the center of gravity of the data set. On the other hand, the robust density based fitness (Fig. 5 (1st row)) succeeds in detecting the correct cluster locations even in the midst of noise.

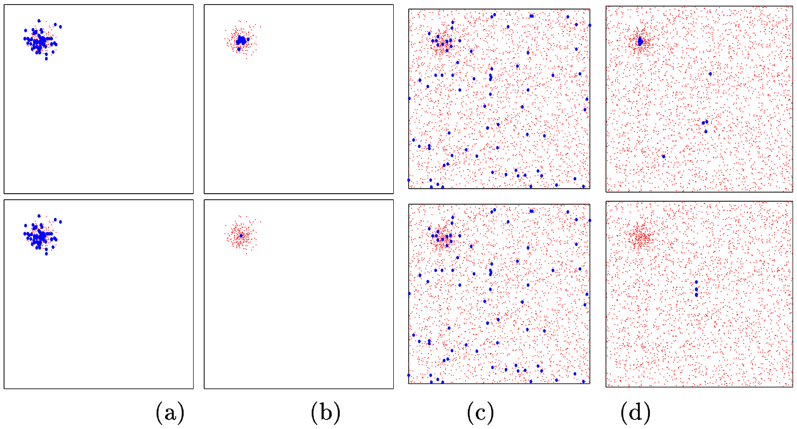


Fig. 5. Evolution of the population using unrefined UNC and density based fitness (1st row) versus Sum of Squared Errors fitness (2nd row): (a) Initial population for clean data set (b) population after 200 generations for clean data set; (c) Initial population for noisy data set (d) population after 200 generations for noisy data set

8 Conclusion

Most existing clustering techniques necessitate the derivation of the optimal prototypes by differentiation to guarantee convergence to a *local* optimum, which can be impossible for most subjective and non-metric dissimilarity measures. For this reason, Evolutionary clustering methods are preferrable in many real world problems. Unfortunately most evolutionary clustering techniques are sensitive to noise, and assume a known numer of clusters. We explained how **(i)** robustness to noise can be achieved with a robust fitness measure, while **(ii)** scalability of the search space with respect to the number of clusters and to the size of the data can be achieved by encoding a single cluster prototype in the chromosome, how **(iii)** the resulting multimodal optimization problem should be solved using a *niching* strategy, which offers the additional advantage of allowing the determination of the number of clusters automatically, and finally that **(iv)** a hybrid Piccard niche size estimation strategy is the key to implement a sound mating restricion, and is the key to successful fitness sharing. Our survey of most existing evolutionary clustering techniques concluded that the only Evolutionary clustering technique that satisfies these requirements is the Unsupervised Niche Clustering (UNC) algorithm. Finally we presented an automatic hybrid Piccard niche size updating strategy which allows fitness sharing to be used within UNC instead of Deterministic Crowding with almost equal success.

Acknowledgment

This work is supported by a National Science Foundation CAREER Award IIS-0133948 to O. Nasraoui.

References

1. R. Duda and P. Hart, *Pattern Classification and Scene Analysis*, Wiley Interscience, NY, 1973.
2. L. O. Hall, I. O. Ozyurt, and J. C. Bezdek, "Clustering with a genetically optimized approach," *IEEE Trans. Evolutionary Computations*, vol. 3, no. 2, pp. 103–112, July 1999.
3. 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.
4. V. V. Raghavan and K. Birchand, "A clustering strategy based on a formalism of the reproductive process in a natural system," in *Second International Conference on Information Storage and Retrieval*, 1979, pp. 10–22.
5. J. N. Bhuyan, V. V. Raghavan, and K. E. Venkatesh, "Genetic algorithms for clustering with an ordered representation," in *Fourth International Conference on Genetic Algorithms*, 1991, pp. 408–415.
6. D. E. Goldberg, *Genetic algorithms in search, optimization, and machine learning*, Addison-Wesley, New York, 1989.
7. G. P. Babu and M. N. Murty, "A near-optimal initial seed value selection in the k-means algorithm using a genetic algorithm," *Pattern Recognition Letters*, vol. 14, no. 10, pp. 763–769, Oct. 1993.
8. D. B. Fogel and P. K. Simpson, "Evolving fuzzy clusters," in *International Conference on Neural networks*, San Francisco, CA, 1993, pp. 1829–1834.
9. 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.
10. C.-Y. Lee and E. K. Antonsson, "Dynamic partitional clustering unsing evolution strategies," in *3rd Asia Pacific Conf. on simulated evolution and learning*, Nagoya, Japan, 2000.
11. P. J. Rousseeuw and A. M. Leroy, *Robust Regression and Outlier Detection*, John Wiley & Sons, New York, 1987.
12. V. Estivill-Castro and J. Yang, "Fast and robust general purpose clustering algorithms," in *Pacific Rim International Conference on Artificial Intelligence*, 2000, pp. 208–218.
13. O. Nasraoui and R. Krishnapuram, "A novel approach to unsupervised robust clustering using genetic niching," in *Ninth IEEE International Conference on Fuzzy Systems*, San Antonio, TX, May 2000, pp. 170–175.
14. S. W. Mahfoud, "Crowding and preselection revisited," in *2nd Conf. Parallel problem Solving from Nature, PPSN '92*, Brussels, Belgium, Sep. 1992.
15. E. Ruspini, "Numerical methods for fuzzy clustering," *Information Science*, vol. 12, pp. 319–350, 1970.
16. P. J. Huber, *Robust Statistics*, John Wiley & Sons, New York, 1981.
17. J. H. Holland, *Adaptation in natural and artificial systems*, MIT Press, 1975.
18. 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.
19. 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.
20. 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, Sep. 1996, pp. 1031–1035.