Spatial Clustering for Data Mining with Genetic Algorithms

Vladimir Estivill-Castro

Neurocomputing Research Centre
Queensland University of Technology,
GPO Box 2434, Brisbane 4001, Australia.
vladimir@fit.qut.edu.au

Alan T. Murray

Australian Housing and Urban Research Institute
Queensland University of Technology,
GPO Box 2434, Brisbane 4001, Australia.
a.murray@qut.edu.au

September 1, 1997

Abstract

Spatial data mining is the discovery of interesting relationships and characteristics that may exist implicitly in spatial databases. The identification of clusters in spatially referenced data provides a means of generalization of the spatial component of the data associated with a Geographical Information System. A variety of clustering formulations exists. A non-hierarchical approach in Data-mining applications is to use a medoid based version. This approach has robust behavior with respect to outliers and many heuristics have been developed that find near optimal partitions. This paper develops a genetic search heuristic for solving medoid based clustering problems. We base our genetic recombination upon Random Assorting Recombination. A comparison is made with previous solution approaches. Results show improvements on the genetic search heuristic.

KEYWORDS: Data Mining, Spatial data sets, Genetic Algorithms, Clustering.

1 Introduction

Geographical Information Systems have served an important role in the creation and manipulation of large spatial databases. Spatial data mining [13] is the discovery of interesting relationships and characteristics that may exist implicitly in spatial databases. The automatic knowledge discovery process in spatial databases aims at a) extracting interesting spatial patterns and features, b) capturing intrinsic relationships between spatial and non-spatial data, c) presenting data regularity concisely and at higher conceptual levels, and d) helping to reorganize spatial databases to accommodate data semantics and to achieve better performance.

Clustering detection in spatially referenced data provides a means of generalization that is complementary to techniques for generalization used in data mining in relational databases [3]. Clustering is the task of identifying groups in a data set based upon some criteria of similarity [4]. Moreover, in geo-referenced space the most obvious measure of similarity is Euclidean distance, although other derived distances are possible. Thus, similarity measurement between gre-referenced database entities is relatively well defined. Clustering, or cluster analysis, has a direct interpretation for knowledge extraction [2, 6, 9, 13, 16].

A variety of clustering formulations exist. A suitable approach for data mining applications is to use a medoid-based optimization formulation [11, 13, 12, 7]. This is due to its robust behavior with respect to outliers and because numerous heuristics find near optimal partitions. The partition of n items into k clusters is achieved by selecting a subset of k items as medoids and assigning every item to its closest medoid. The most common heuristics are a form of "hill-climbing" that guarantees local optimality. However, this is a domain where the objective function has many local optima and where genetic algorithms may prove to be capable of producing superior solutions.

We implement a clustering method using genetic algorithms for solving the medoids based formulation of clustering. We base our genetic recombination on Random Assorting Recombination [14]. This provides desirable properties (respect and proper assortment) to the generic search. Our genetic operators and design take into consideration that, for spatial data-mining applications, data sets are large while the number of clusters is typically small.

The medoid-based formulation of clustering is a special case of a well-studied problem known as the p-median problem. There have been many proposals and approaches for solving this problem, both heuristically and exactly. Hosage and Goodchild [10] applied a binary encoded genetic algorithm and found this approach unlikely to compete with existing solution approaches. However, Bianchi and Church [1] found that a non-binary encoding resulted in a much superior genetic search and, in some aspects, it was competitive with existing approaches. Both of these previous uses of genetic algorithms are based on simple crossover, mutation and inversion. The main difference is the encoding as binary strings or as integer strings. Because of the genetic operators used, they both face problems with the creation of infeasible solutions and the introduction of physical bias. Our approach is an improvement in both aspects.

Other attempts to use evolutionary techniques for spatial clustering [8, 9] are less suitable for spatial data mining applications since they are designed for only small data sets given the delicate encoding and shape restrictions on the clusters. Namely, each cluster requires several bits to encode the parameters of the neighborhood (centroid and radius, or other parameters). Further, clusters are limited to the shape of ellipsoids or squares, limiting the applicability of the methods and certainly ignoring issues like outliers.

Our approach to the application of genetic algorithms for the clustering problem is to use operators that combine sets [15, 14]. This provides a real-world illustration of the usefulness of Random Assorting Recombination. Moreover, we provide a more efficient implementation for Random Assorting Recombination than its early description [15, 14]. As a result we obtain an improved crossover operator that balances respect and assortment.

2 Medoid based clustering

Formulations of clustering problems vary by the criterion that measures the quality of the partition and usually correspond to some evaluation of the within group difference or the cohesion within items in a cluster versus the distinctness among clusters. For the large number of observations that Knowledge Discovery applications are pursuing, obtaining optimal solutions for formulations of clustering is unrealistic because these problems are NP-complete. However, certain approaches do have advantages over others, and in particular, the medoids approach offers robustness with respect to outliers as well as a structure which can be solved approximately by several techniques [7, 12]. We now describe the medoids approach.

Consider a set of data items $P = \{p_1, p_2, \ldots, p_n\}$ where each p_i is a point in d-dimensional real space $\Re^d$. The clustering problem consists of naturally grouping these points into k clusters. A common approach to clustering is to identify a representative for each cluster and asses the quality of the clustering as the average distance between items and their representative. In the medoids approach, the set R of representatives is restricted to be a subset of P. Thus, the clustering problem translates to a combinatorial optimization problem where the goal is to find a set of representatives that minimizes the following criterion F defined by

$$F(R) = \sum_{i=1}^{n} d(p_i, rep[p_i, R])$$

over all subsets R of P with $\|R\| = k$. Usually, the distance d corresponds to a Minkowski distance (i.e. $d_p(\vec{x}, \vec{y}) = (\sum_{u=1}^{d} |x_u - y_u|^p)^{1/p}$). If $rep[p_i, R]$ is the closest point in $R = \{m_1, \ldots, m_k\}$ to p_i, then $\min_{j \in \{1,\ldots,k\}} d(p_i, m_j) = d(p_i, rep[p_i, R])$.

The search space is the set $X \subset 2^P$ of all sets $R \subseteq P$ with $\|R\| = k$. The objective function F is a function $F : X \rightarrow \Re$ that to each subset $R \subseteq P$ assigns a real value $F(R)$. The value $F(R)$ represents the quality of the clustering.

Figure 1 illustrates the medoids approach for a data set of 12 bidimensional points. The two plots show the same partition into two clusters. However, the representative of the left cluster is different, and thus, the function F is smaller for the right plot than for the left plot. In practice, the size of X is large enough that exhaustive search is infeasible even for relatively small problems. In this paper we use genetic search for optimizing F.

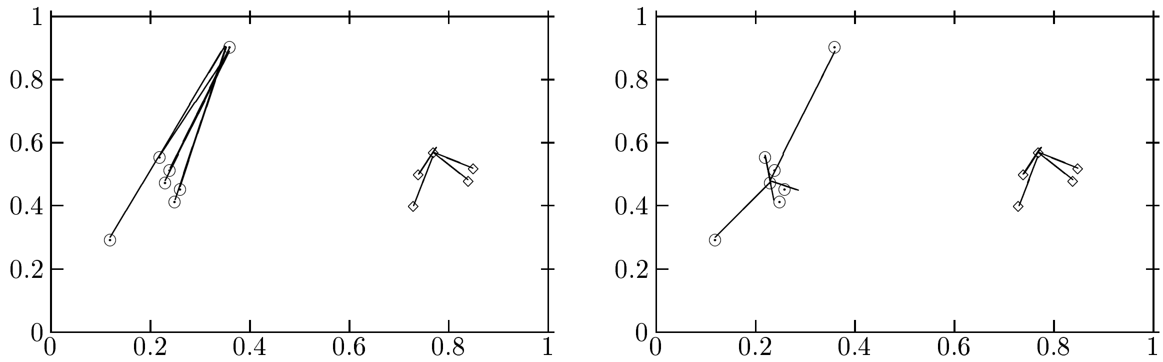

Figure 1: Two examples of Medoid clusterings.

3 Searching for a set of k medoids

Genetic Algorithms are an optimization technique that has been shown to be robust for a variety of complex search spaces. The technique offers a trade-off of exploration and exploitation by maintaining a fixed-size population of chromosomes. Typically, each chromosome encodes a feasible solution. Iteratively, a new population replaces a previous one. The new population results from probabilistic selection of parental chromosomes whose combination produces offspring. The parents are selected randomly but in proportion to their relative merit as a solution. Recombination operators are the mechanisms for producing offspring.

For our clustering problem, a feasible solution is a set R of fixed size k. The choice of an encoding mechanism as well as a recombination operator with adequate characteristics can be difficult [14, 15]. Traditional genetic search encodes chromosomes as binary strings. This approach would lead to a representation for a subset R of P in terms of its characteristic vector. That is, a canonical order would be established in $P = \{p_1, \ldots, p_n\}$. The i-th position in the binary string would be 1 if $p_i \in R$ or 0 if $p_i \notin R$. This has many drawbacks for clustering in applications with $\|R\| << \|P\|$, one of which is waste of computer memory. Other problems are ensuring that traditional recombination operators produce offspring with exactly k bits set to 1 [10] (that is, a feasible solution). We encode a solution with an integer string as done in Bianchi and Church [1]. That is, we implement each chromosome as an array C of k different integers in $[1, n]$. This encodes the set $R \subset P$ by the rule $p_i \in R$ if and only if $\exists\ j$ such that $C[j] = i$.

Along with the choice of encoding is the design of its recombination operators. The use of integer string encoding can be problematic in that traditional genetic operators may produce infeasible solutions. For example, *simple crossover* potentially produces integer arrays with one or more repeated values. This enlarges the search space. The filtering of infeasible solutions adds computational overhead. The recombination operators should ideally offer

- reduced physical bias [5],
- respect [14], and

- proper assortment [14].

Physical bias is the non-uniform preference and production of offspring due to how chromosomes are encoded and irrespective of what the encoding represents. Ideally, the potential alleles (or subspaces of the search space) for which equivalent information has been collected should be equally likely. At least, the sampling probability of a search region should be as independent as possible of the position of alleles in the encoding. A respectful operator preserves common characteristics of parents in produced offspring. Proper assortment means that all combinations of compatible characteristics present in the parents most have a non-zero probability of being present in the offspring.

We seek a subset R of P such that $F(R)$ is a minimum. The most simple relation between a set and an item is membership. Thus, a characteristic of a candidate solution R is that it contains a point p. If two solutions share a characteristics, this means two solutions R_1 and R_2 share a point $p_i \in P$. A respectful recombination operator of R_1 and R_2 would always produce offspring that includes all points in $R_1 \cap R_2$. Assortment means that if a characteristic is present in R_1 ($p_i \in R_1$ but perhaps $p_i \notin R_2$) and a characteristic is present in R_2 ($p_j \in R_2$ but perhaps $p_j \notin R_1$), then it is possible to produce offspring R_o with $\{p_i, p_j\} \subset R_o$.

Characteristics like R contains p_i or R does not contain p_i can be formalized as equivalence relations in the search space $X = \{R \mid R \subset P \ \wedge \ \|R\| = k\}$. Radcliffe [14] has shown that these characteristics form an orthogonal basis up to level k of a general set of equivalence relations and that respect and proper assortment is unattainable. The Random Respectful Recombination ($\mathcal{R}^3$) operator allows respect but a weak level of assortment while Random Assorting Recombination ($\mathcal{RAR}_\omega$) uses a parameter ω to improve assortment for a regulated penalty in respect [14].

The binary recombination operator $\mathcal{R}^3$ is conceptually simple. When combining two sets R_1 and R_2 of size k, all elements in $R_1 \cap R_2$ are offspring members. The remaining $k - \|R_1 \cap R_2\|$ places are randomly selected from unused elements of the two parents R_1 and R_2. More formally, given two sets R_1 and R_2, the operator $\mathcal{R}^3$ chooses randomly and uniformly the offspring from the set $\{R \mid R \subset P, \|R\| = k, R_1 \cap R_2 \subseteq R\}$. Efficient implementation of $\mathcal{R}^3$ is delicate, especially in the case $k << n$.

Similarly, efficient implementation of $\mathcal{RAR}_\omega$ is not simple. The original algorithms by Radcliffe and George [15] demand the manipulation of “barred elements” to indicate their absence in both parents. There are $O(n-k) = O(n)$ such barred elements and thus, the original algorithms for $\mathcal{RAR}_\omega$ are costly when $k << n$. Thus, we have redesigned the implementation of $\mathcal{RAR}_\omega$ ensuring that the operator tends to $\mathcal{R}^3$ when $\omega \to \infty$ and that has stronger assortment for small values of ω.

Our implementation of $\mathcal{RAR}_\omega$ works as follows. Given R_1 and R_2 as parents, the offspring R_o is built iteratively, adding one point at a time until it has size k. We chose point p for inclusion into R_o by drawing uniformly a random number ρ in $[0, 1]$ and comparing it with a fractional value, *cut* (*cut* will be defined shortly). If $\rho \leq cut$, then a point p is selected randomly and uniformly in $R_1 \cap R_2 - R_o$. If $\rho > cut$, then the point p is selected randomly and uniformly in $(R_1 \cup R_2) - [(R_1 \cap R_2) \cup R_o]$. In every case, R_o is updated by $R_o \leftarrow R_o \cup \{p\}$.

If $R_1 \cap R_2 - R_o$ or $(R_1 \cup R_2) - [(R_1 \cap R_2) \cup R_o]$ is empty before R_o has k elements, then R_0 is completed with random elements from $(R_1 \cup R_2) - R_o$.

We define $cut = 1 - 1/\omega$. We have found that, in particular, $cut = 2/3$ ($\omega = 3$) works well. Refinement to particular problem instances is often necessary.

The capacity of handling subsets of size k from a large universe has been accomplished here. Our implementation of $\mathcal{RAR}_\omega$ preserves the feature that for a fixed ω the level of positive assortment is adaptive to the similarity between parents. This is important, since as the genetic algorithm converges, the genetic diversity is reduced and parents look much more alike. The genetic algorithm needs strong assortment in later generations to preserve its exploration capability.

Note that although our encoding has some redundancy (the same set can be represented by any permutation of the array entries), our genetic operators are based on a logical structure (the phenotype) and not on the physical structure (the genotype). The significance is that there is no physical bias and the search space is not enlarged by the encoding.

Other details of our genetic algorithm are as follows. Selection is by roulette proportional to relative fitness and population size remains constant through all generations. The previous generation is fully replaced by new offspring. Finally, there is a mutation operator that with very low probability can modify a set R by swapping $p_i \in R$ with $p_j \notin R$.

4 A comparison

In order to perform a comparison we have generated test data by selecting k points $\vec{c}_1, \ldots \vec{c}_k$ randomly and uniformly in $[0,1] \times [0,1]$. These points represent virtual centers and are not present in the actual data set. The smallest separation $D = \min_{i \neq j} d(\vec{c}_i, \vec{c}_j)$ is used to determine a common virtual radius $r = D/2$. With this information, as many as $(1 - N)n$ data points are chosen by first selecting randomly a virtual centroid $\vec{c}_i$ and then randomly selecting polar coordinates γ_i, ψ_i (γ_i is uniform in $[0, r]$ and ψ_i is uniform in $[0, 2\pi]$). The point in the data set is $\vec{c}_i + (\gamma_i \sin \psi_i, \gamma_i \cos \psi_i)$. The data set is shuffled with Nn randomly and uniformly selected points where $N \in [0,1]$ is a percentage of the amount of noise. Figure 2 (a) shows a data set generated using $n = 100$, $k = 10$ and $N = 10$. The virtual centers are shown using $+$.

Figure 2 (b) shows a clustering obtained with the well-known statistical method called k-means [4] which differs from the medoid approach discussed previously. The centers found by k-means are indicated using $\odot$. Note that k-means merges three far apart clusters into one. High quality solutions to F, may be found using hill-climbing approaches [7]. The Local Hill climbing [7] approach applied 20 times to the sample data set with different starting points results in the solution of Figure 3 (a). This discovers 9 of the 10 clusters, and only places one medoid on an outlier. There are two reasons for this. First, from the point of view of criterion F, a solution with 9 clusters is superior to the solution with 10 clusters shown in Figure 3 (b). Second, two virtual centers happen to be so close that a statistical test favors the use of 9 groups rather than 10. The application of our genetic algorithm identified two different clusterings. The first was equivalent to that given in Figure 3 (a). The second is the

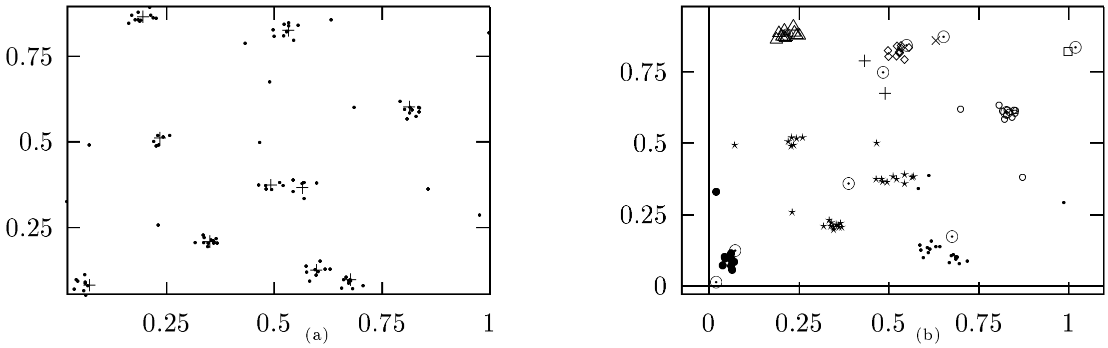

Figure 2: A data set of 100 points clustereding with k-means.

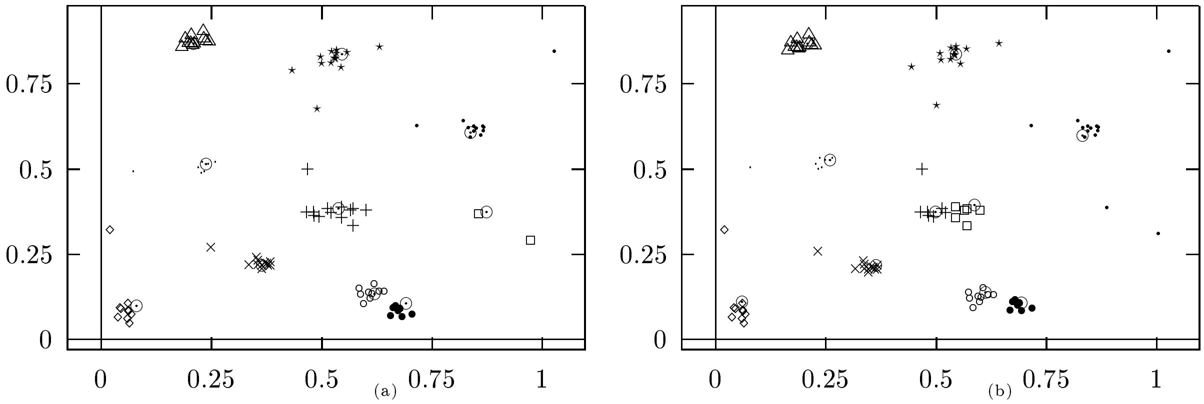

Figure 3: Clustering with Local Hill climbing and with the $\mathcal{RAR}_\omega$-GA.

Table 1: A comparison of search approaches.

20 independent runs			
	Local Hill climbing	Traditional Recombination	$\mathcal{RAR}_\omega$ recombination
9 clusters	20 times	15 times	19 times
10 clusters	0 times	5 times	1 time

Table 2: Minimizing F with two GA approaches.

Value of F averaged over 20 independent runs	
Traditional Recombination	$\mathcal{RAR}_\omega$ recombination
3.748	3.715

solution given in Figure 3 (b). Interestingly, the Figure 3 (b) corresponds to a solution with the 10 original clusters, however, this solution is suboptimal solution with respect to criterion F. Table 1 shows how often a suboptimal solution with respect to criterion F was produced with the traditional operators approach [1] and with our approach. The genetic algorithms were run for 100 generations with a population size of 50. All other parameters and selection criteria were the same except for $\omega = 3$ in $\mathcal{RAR}_\omega$ and the probability of mutation to 0.01. Clearly, the $\mathcal{RAR}_\omega$ approach proposed here is an improvement over the genetic algorithm using traditional operators [1, 10].

Moreover, we computed the average objective function value F for the 20 solutions found. Table 2 shows that the $\mathcal{RAR}_\omega$ approach provides an improvement over traditional operators in this respect as well. For this data set, the optimal value of F is 3.466. Thus, a lower average is superior. However, this is not surprising given the results in Table 1.

5 An illustration

Clustering is a mechanism for generalization that is central to Knowledge Discovery. To illustrate the role of spatial clustering in Data Mining we present a constructed example. Let us assume that Figure 4 (a) is the location in some urban area of 3 types of crimes (stolen vehicles, break-ins, and robberies). Figure 5 (a) is the location of churches, Figure 5 (b) is the location of parks, and Figure 5 (c) is the location of the subway stations.

The mining agent may explore the data in an attempt to find a rule or some link associated with the occurrence of stolen cars. Thus, locations of stolen vehicles is highlighted in Figure 4 (b). These highlighted points are equivalent to those given in Figure 2 (a). Thus, applying the $\mathcal{RAR}_\omega$-GA results in a clustering such as that given in Figure 3 (b). The medoids of this clustering are illustrated in Figure 6 (a). The mining agent then will

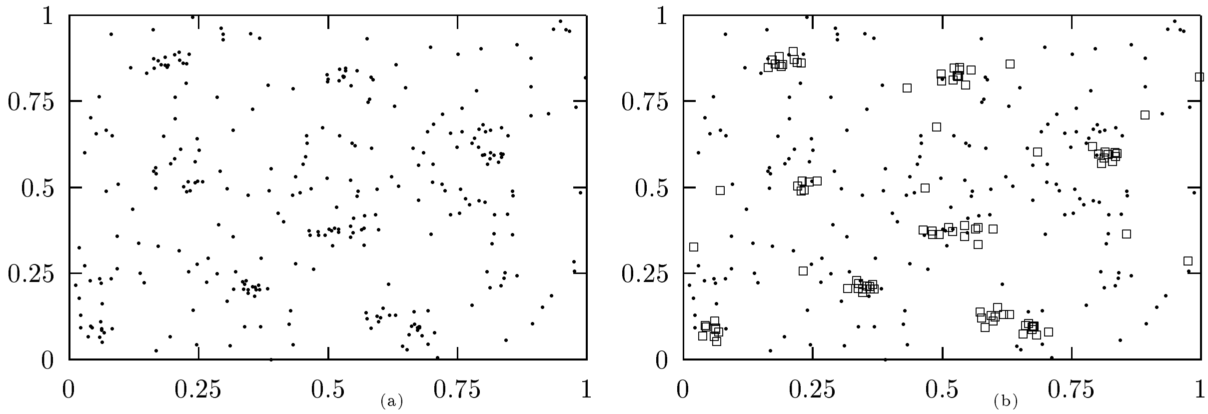

Figure 4: (a) crimes occurrences and (b) highlighted location of stolen vehicles.

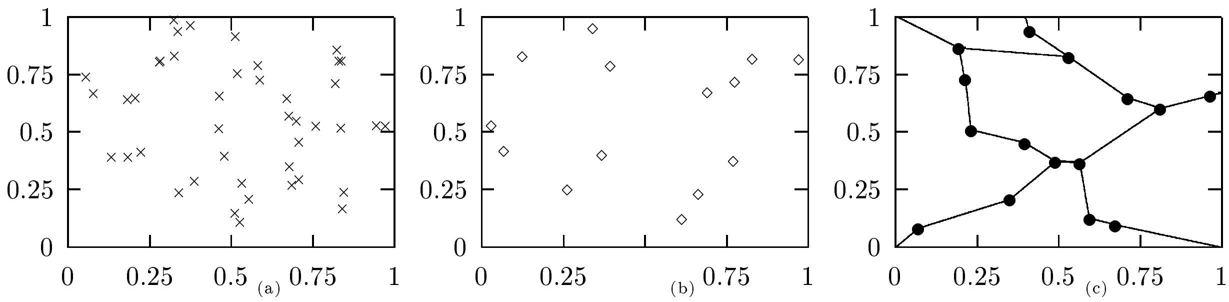

Figure 5: (a) Location of churches, (b) centroids of parks, and (c) subway lines with subway stations.

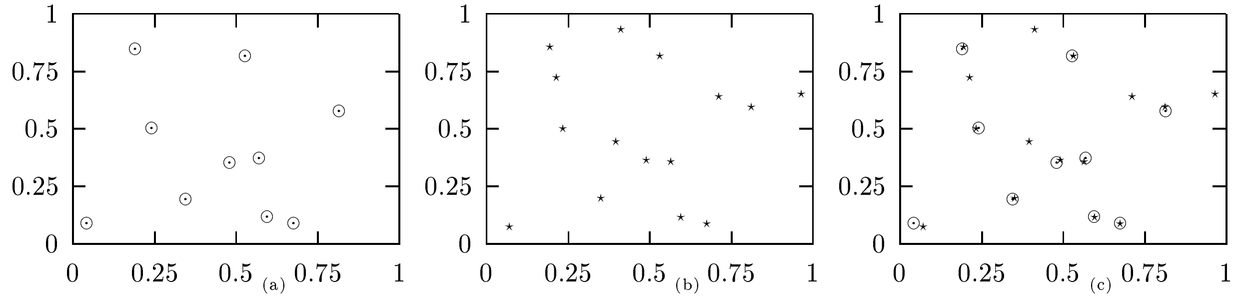

Figure 6: (a) Medoids with $\mathcal{RAR}_\omega$-GA, (b) subway stations, and (c) overlay of the two.

overlay the medoids with each of the other 3 thematic data layers of Figure 5. By computing the average of the distances form medoids to closest point in each layer, the agent finds that the distance in unusually small when the medoids and the train stations are overlaid (refer to Figure 6(a)-(c)). More precisely, consider n_1 the number of points in layer $L_1 = \{p_1, \ldots, p_{n_1}\}$ and n_2 the number of points in layer $L_2 = \{q_1, \ldots, q_{n_2}\}$ (without loss of generality assume that $n_1 \leq n_2$). Let $rep[p_i, L_2]$ be the closest point to p_i in L_2. If $S(L_1, L_2) = \sum_{i=1}^{n_1} rep[p_i, L_2]/n_1$ or $M(L_1, L_2) = \max_{i=1,\ldots,n_1} rep[p_i, L_2]/n_1$ are small and far from their expected values (assuming all p_i and q_i are uniformly and independently distributed), the miner signals a potential relationship. In our example, the $S(\mathcal{RAR}_\omega medoids, stations)$ is 0.011 and $M(\mathcal{RAR}_\omega medoids, stations)$ is 0.033 while the 95% confidence interval of the expected value of $S(10, 15)$ is 0.13 ± 0.007 and the expected value of $M(10, 15)$ is 0.27 ± 0.023. This allows the miner to discover a relationship between the location of stolen cars and subway stations.

6 Final remarks

Using set recombination with $\mathcal{RAR}_\omega$ provided an improvement over the traditional crossover, mutation and inversion. The results of our preliminary experimentation showed equivalent CPU-time requirements using traditional operators or set recombination. Genetic algorithms remain slow in comparison to Local Hill climbing.

The theoretical properties of $\mathcal{RAR}_\omega$ suggest that this line of work for genetic algorithms deserves further investigation. Issues like elitism and ranking versus roulette selection may provide further improvements. Our results open the possibility for even more competitive genetic search for medoid based clustering.

7 Acknowledgments

This research was supported in part by a grant from the Australian Research Council.

References

[1] G. Bianchi and R. Church. A non-binary encoded genetic algorithm for a facility location problem. Working Paper, 1992. Department of Geography, University of California, Santa Barbara.

[2] T. Brinkhoff and H.P. Kriegel. The impact of global clustering on spatial database systems. In J. Bocca, M. Jarke, and C. Zaniolo, editors, *Proceedings of the 20th Conference on Very Large Data Bases (VLDB)*, pages 168–179, San Francisco, CA, 1994. Santiago, Chile, Morgan Kaufmann Publishers.

[3] Y. Cai, N. Cercone, and J. Han. Attribute-oriented induction in relational databases. In G. Piatetsky-Shapiro and W.J. Frawley, editors, *Knowledge Discovery in Databases*, pages 213–228, Menlo Park, CA. USA, 1991. AAAI Press.

[4] R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York, US, 1973.

[5] L.J. Eshelman, R.A. Caruana, and J.D. Schaffer. Biases in the crossover landscape. In J.D. Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*, pages 10–19, San Mateo, CA., 1989. George Mason University, Morgan Kaufmann Publishers.

[6] M. Ester, H.P. Kriegel, S. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In E. Simoudis, J. Han, and U. Fayyad, editors, *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pages 226–231, Menlo Park, CA, 1996. AAAI, AAAI Press.

[7] V. Estivill-Castro and A.T. Murray. Mining spatial data via clustering. Technical Report #5/97, Faculty of Information Technology, Queensland University of Technology, Brisbane 4000, Queensland, Australia, 1997.

[8] D.B. Fogel and P.K. Simpson. Evolving fuzzy clusters. In *Proceedings of the 1993 IEEE Conference on Neural Networks*, pages 1829–1834, Piscataway, NJ, 1993. IEEE, IEEE Press.

[9] A. Ghozeil and D.B. Fogel. Discovering patterns in spatial data using evolutionary programming. In J.R. Koza, editor, *Genetic Programming: Proceedings of the First Annual Conference*, pages 521–527, Cambridge, MA, 1996. MIT Press.

[10] C. Hosage and M. Goodchild. Discrete space location-allocation solutions from genetic algorithms. *Annals of Operations Research*, 6(1):35–46, 1986.

[11] L. Kaufman and P.J. Rousseuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley & Sons, NY, US, 1990.

[12] A.T. Murray and V. Estivill-Castro. Cluster discovery techniques for exploratory spatial data analysis. Submitted manuscript, 1997.

[13] R.T. Ng and J. Han. Efficient and effective clustering methods for spatial data mining. In J. Bocca, M. Jarke, and C. Zaniolo, editors, *Proceedings of the 20th Conference on Very Large Data Bases (VLDB)*, pages 144–155, San Francisco, CA, 1994. Santiago, Chile, Morgan Kaufmann Publishers.

[14] N.J. Radcliffe. Genetic set recombination. In L. D. Whitley, editor, *Foundations of Genetic Algorithms 2*, pages 203–219, San Mateo, CA, 1993. FOGA-92 Second Workshop on the Foundations of Genetic Algorithms and Classifier Systems, Morgan Kaufmann Publishers.

[15] N.J. Radcliffe and F.A.W. George. A study of set recombination. In S. Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 23–30, San Mateo, CA, 1993. University of Illinois at Urbana-Champaign, Morgan Kaufmann Publishers.

[16] T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH:an efficient data clustering method for very large databases. *SIGMOD Record*, 25(2):103–114, June 1996. Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data.