Strategies for an Artificial Life System to cluster high dimensional Data

Alfred Ultsch
Databionics Research Group
University of Marburg, D-35032 Marburg, Germany
ultsch@informatik.uni-marburg.de

Systems for clustering with collectives of autonomous agents follow either the ant approach of picking up and dropping objects or the DataBot approach of identifying the data points with artificial life creatures. In DataBot systems the clustering behaviour is controlled by movement programs. This paper reports the answers to two questions regarding movement: first, what is the most elementary movement program for clustering and second, what movement strategy can effectively solve even very difficult cluster problems. The clustering abilities are tested on synthetic data which are difficult to cluster. The effective movement strategy found is applied to real world data of stock markets and DNA microarrays.

1. Introduction

Ant colony simulations for the clustering of objects have been published, for example, by Bonabeau et al [1]. Such systems imitate how ants organize their brood and cemeteries. Lumer and Faieta applied this approach to exploratory data mining [2]. Contemporary ant systems, like the stigmergic swarm systems of Ramos and Abraham, are reported to cluster data sets of interesting sizes, such as in world wide web mining applications [3]. In ant simulation systems data points are represented as objects which are picked up and dropped. The behaviour of the ants is modelled according to the observed behaviour of biological ants. We proposed 1999 independently a system for clustering with a population of autonomous agents, so called DataBots [4,5]. In DataBot systems the data is identified with the moving Artificial Life creatures. Self organization into clusters depends in these systems on the movement programs. This paper reports the answers to two questions regarding movement in DataBot systems: first, what is the most elementary movement program for clustering and second, what movement strategy can effectively and efficiently solve even very difficult cluster problems.

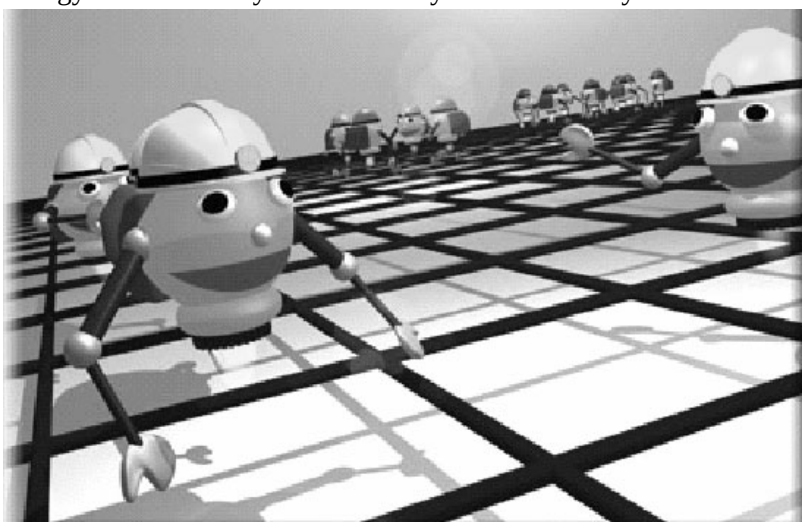

Figure 1: Artist's view of DataBots

The remainder of the paper is organized as follows: chapter 2 gives a short description of the DataBot approach. Chapter 3 expains the visualisation tools for clustering with DataBots. A mathematical basis for short- vs long range interaction is outlined in chapter 4. Chapter 5 reports the minimal strategy for clustering. In chapter 6 synthetic data sets are described which are difficult to cluster. Chapter 7 reports the efficient movement strategy that effectively clustered all our data sets. In Chapter 8 an application of the DataBot system to stock marked analysis and prediction is presented. Chapter 9 describes the application of the DataBot clustering to a difficult high dimensional problem in Bioinformatics: cDNA microarrays.

2. The DataBots system

Let $D \subset R^n$ be the subspace of R^n where data points can be observed in principle. A data set is a finite set $E = \{x_1, ... x_d\}$ with $x_i \in D$. A distance measure is defined on the data space $D \times D \rightarrow R+$: x, $y \in D$, $d(x,y) \geq 0$. More similar pairs of E have smaller values of d. For each data point an associated DataBot is created. A DataBot is an autonomous agent moving on an UD-Matrix (see below). Sensors of DataBots are eye and nose. With their eyes DataBots are able to sense distances and directions to other DataBots. Noses are used to take in and identify pheromones. Pheromones are emitted by DataBots. The pheromones are the high dimensional data points x_i associated with the particular DataBot DB_i. DataBots dwell on a UD-Matrix. A UD-Matrix is a set of nodes connected by a rectangular grid of paths. The UD-Matrix is finite but borderless using a toroid structure of the grid layout. See Figure 2. Each node is connected to its neighbours in four directions (see Figure 3).

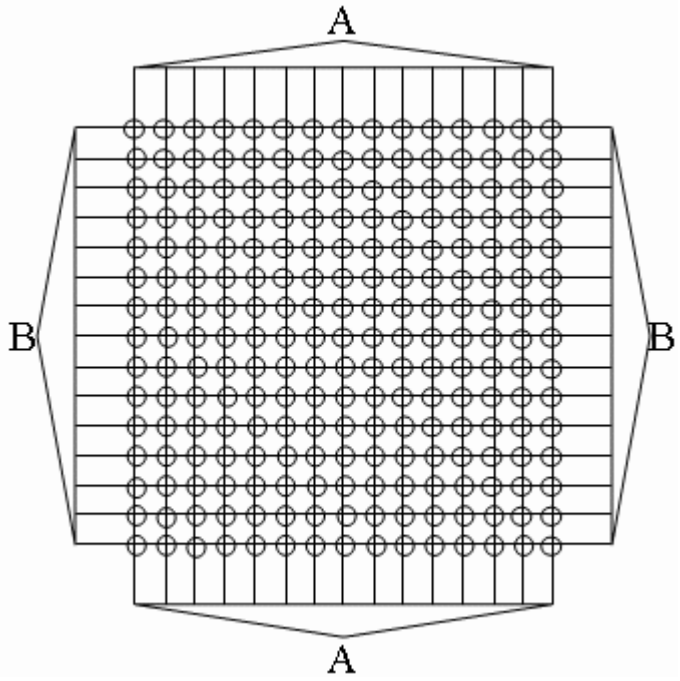

Figure 2: Toroid structure of the UD-Matrix,

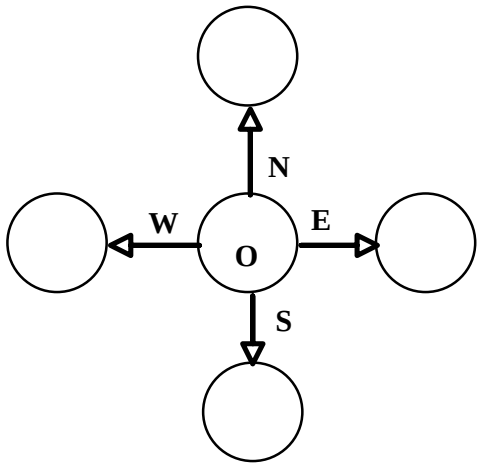

Figure 3: Local connections on the UD-Matrix

We call the directions North, East, South and West. The node itself is called O. Each node may carry and transmit a pheromone. The pheromones of a node are transmitted in each time step to the four neighbouring nodes. The pheromones received from the neighbours are mixed by averaging. At each time step some portion of the pheromone disappears, i.e. the data vector is shortened by a multiplicative constant. At each time step a DataBot may move from the node it sits on (O) to one of

the immediate neighbouring nodes of O, i.e. to the N, E, S, W node. Another possibility is to rest on O. Movement is considered as a stochastic process determined by the five directional probabilities {p(O), p(N), p(E), p(S), p (W)} = PM, with p(X) = p(DataBot has moved to node X). A movement program is an assignment of probabilities to PM. Movement programs may be combined to form movement strategies. Movement strategies consist of weighted summation and rescaling of movement programs such that the resulting PM is a probability distribution. Movement strategies are parameters of the DataBot simulation program ALF implemented by Dirk Malorny [6]. The strategies are formulated in plain text as calls to movement programs. Movement programs are functions implemented in C++. Table 1 shows a movement strategy based on the movement programs “randomStep”, “persistency” and “bestNormalizedMix”. The weighting factor of the programs is given by the first parameter of the movement program calls.

```
randomStep(3.0);
persistency(30.0, 10.0);
bestNormalizedMix(100.0);
```

Table 1: Formulation of a movement strategy for DataBots called “ClusterTendency”

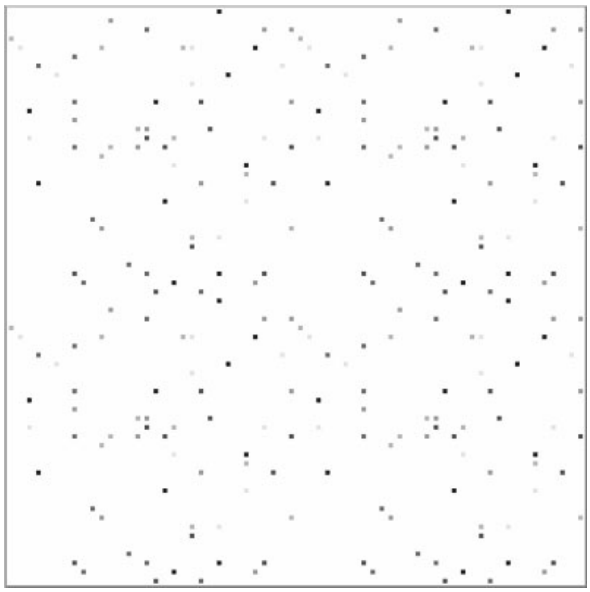

Figure 4: Tiled display of DataBots

3. Visualization of Clustering with DataBots.

DataBots are visualized in ALF as coloured pixels. Colours of DataBots are constructed from a known classification of the data or may be assigned manually. If no prior classification is known, the pheromones are mapped to the RGB colour space, in such a way, that DataBots with similar pheromones are given similar colours. See Figure 4 for a snapshot of a simulation with mapped colours. A tiled display, as proposed in [7], has been used in order to visualize the borderless structure of the toroid UD-Matrix to form a two dimensional display. The UD-Matrix is shown four times in adjacent tiles. This allows a coherent view of emergent structures. Each DataBot is displayed four times on the tiled display. Figure 4 shows this quadrupling of the DataBots representations.

The propagation of the pheromones can be observed by a display of the length of the corresponding data vector at each node. Figure 5 is such a display.

A U-Matrix, as known within the context of self-organizing feature maps (SOM) [9], can be used to observe the formation of clusters in terms of the distances of the data points in input space. After the final position of a DataBot has been reached, the U-Matrix is calculated like in a SOM [8]. During this calculation, however, the positions of the images of the data (pheromones), in SOM terminology called bestmatches, are fixed to the final positions of the DataBots [6]. A U-Matrix is an important tool to judge whether there are clusters present when exploring data sets of unknown structure[7].

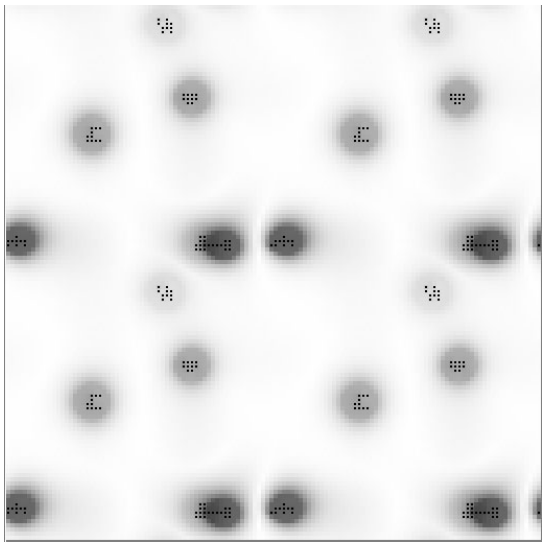

Figure 5: Propagation of pheromones

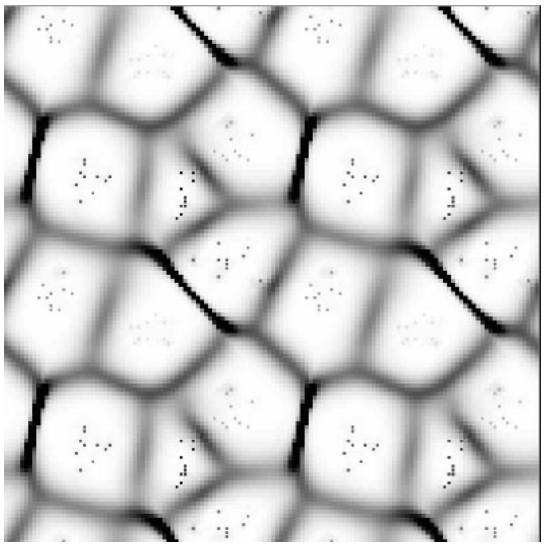

Figure 6: U-Matrix clusters of Hexa data

Figure 6 shows a U-Matrix of the Hexa data as a tiled display.

4. Friends and Foes

A very general requirement of the emergence of structures in self organizing systems seems to us the cooperation in a short range (in the neighbourhood) and concurrency on a long range (afar). This principle is, for example, realized in SOM using a “mexican hat” shaped neighbourhood [9]. To realize this principle it is essential to define, what is in the neighbourhood and what is afar. We follow here the idea of information optimal sets and its application to ABC analysis as presented in [10]. An information-optimal set is minimal in size and produces as much information (entropy) as possible [7]. For subset set sizes measured in fractions of the total set p, size the entropy I(p) can be calculated in percent of the maximal information as: I(p) = -e p ln(p). Define the unrealized potential URP(S) of a set S as the distance from the point (0,1), i.e. the minimal set size producing maximum information. Minimizing URP results in an information optimal set with size u = 20.13%. This set size produces 88% information. The optimality of this set at about (20%, 80%) may serve as an explanation for the so called "Pareto 80/20 law", which is empirically found in many domains [10].

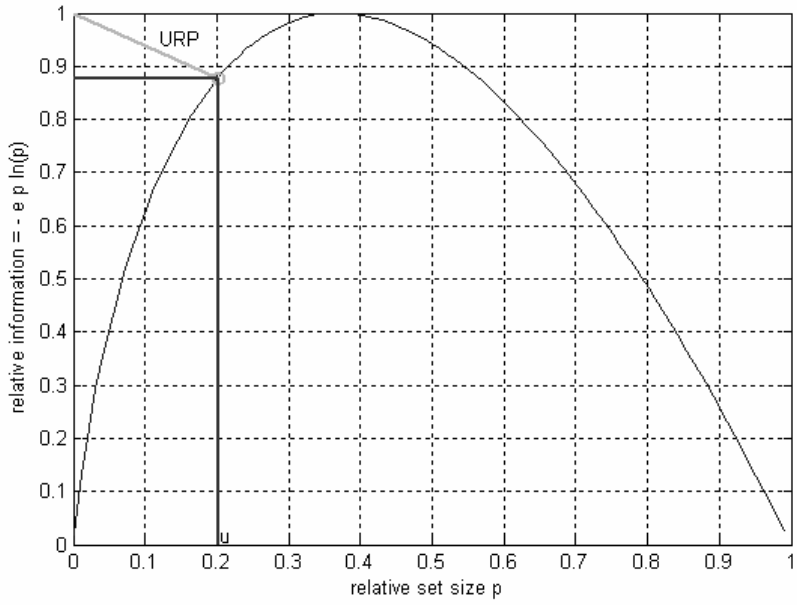

Figure 7: Relative Entropy, URP and the ParetoPoint u

Following this Ansatz a precise formulation of neighbourhood (friends) and distant points (foes) can be derived. The so called ABC-analysis, in marketing and economics classifies customers according to cost and yield into three classes. In [10] the concept of information optimal sets and its relation to ABC-analysis is introduced. In DataBot distances the A class are the friends, the C class the foes. The points in the neighbourhood, i.e. in a subset of size u, can also be used to calculate densities in the high dimensional data set and visualized in form of a P-Matrix [7]. The concept of friends and foes plays an important role for successful movement strategies for DataBots (see below).

5. A minimal strategy for clustering with DataBots

The data set "Hexa" shown in Figure 8 consists of six natural clusters of 10 points. The clusters span the coordinate axis of the coordinates axes of R^3 and are well separated. This data set may serve as the control experiment of any cluster algorithm. Any clustering algorithm must be able to cluster Hexa.

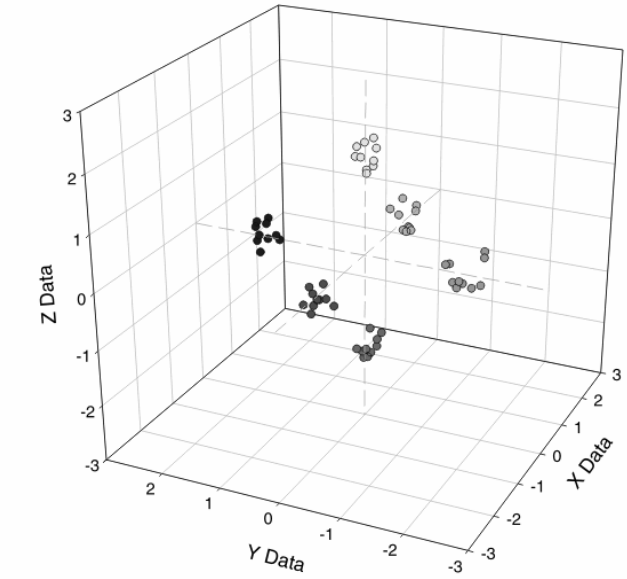

Figure 8: The Hexa data set

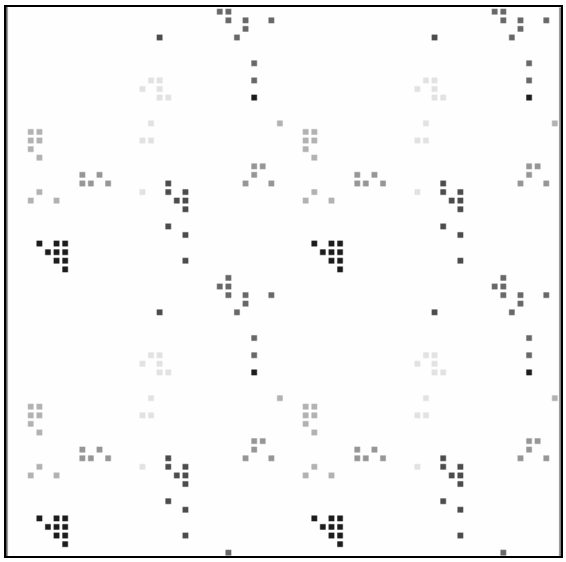

Figure 9: ClusterTendency applied to Hexa

The so called "BestMix" strategy lets a DataBot move with increased probability in direction of the pheromone which is closest to the DataBots own pheromone. Simulations of this movement strategy led to no clustering of Hexa. In particular the DataBots were observed to move back to their previous location since their own pheromone is perceived. Increasing the probability of pursuing a chosen movement direction (the "Persistency" movement program) did not improve clustering. Adding some random walk (the "Random" movement program) results in the "ClusterTendency" strategy formulated in Table 1. As shown in Figure 9, this strategy shows some tendency for local concentrations of similar data points, but no real clustering is achieved [6].

The movement program "FleeWorstFoe" increases the probability for that direction opposite to the most distant pheromone in the DataBots field of vision. A movement strategy which combined the Random, Persisteny, BestMix, and FleeWorstFoe movement programs into "MinimalStrategy" was the simplest movement strategy we found to show a clustering for Hexa [6]. The omission of one of the movement programs from this strategy inhibited the formation of clusters. In this sense MinimalStrategy is a minimal movement strategy for clustering with DataBots.

6. Data sets that are difficult to cluster

The data set called "Lsun" is shown in Figure 10, the data set called "TwoDiamonds" in Figure 11. Lsun is problematic for clustering algorithms for the different shapes, inner cluster variances and small cluster separations. At the touching points of the two diamonds in TwoDiamonds the cluster structure

is defined rather by a change in density than by the points' distances. All the data sets mentioned so far are correctly clustered by DataBots using the FriendsandFoes strategy.

The Chainlink data set has been used in [11] to show that the clustering abilities of emergent SOM are different from k-means clustering. The clusters of three dimensional data set are clearly defined and well separated (see Figure 12). The clusters can, however, never be separated by a linear discrimination function. Cluster algorithms like k-means, Ward and many others are unable to cluster this data set correctly, even if the algorithms are given the information that there are two clusters.

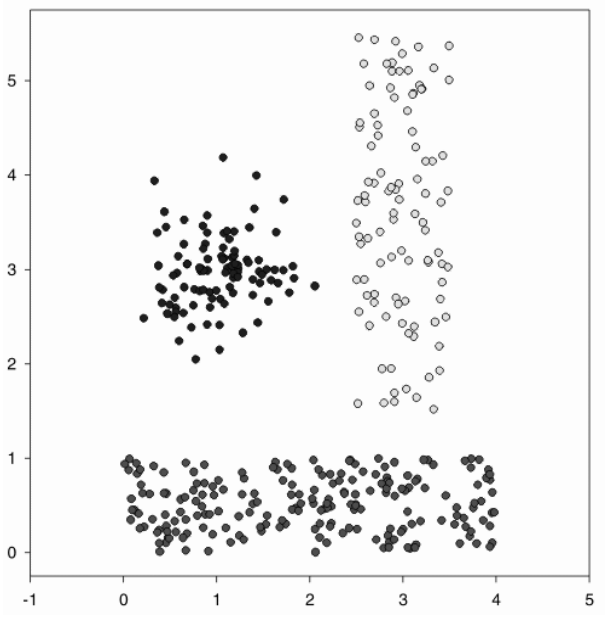

Figure 10: The Lsun data set

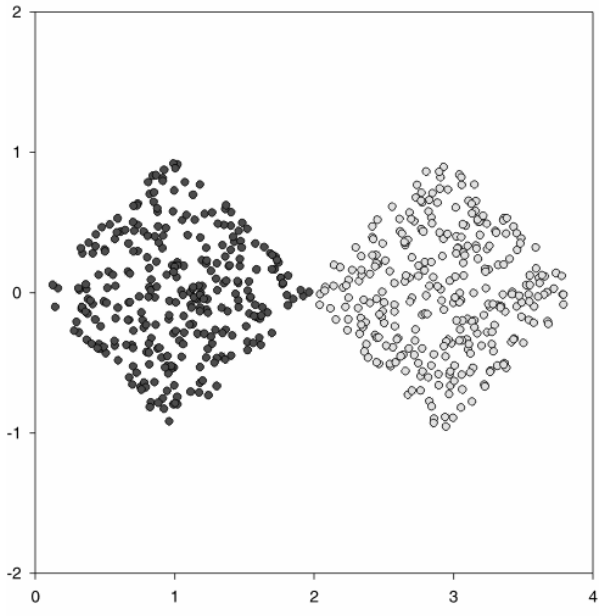

Figure 11: The TwoDiamonds data set

Another difficult data set for clustering is called "Atom" (Figure 13). It consists of 400 data points in the "core" and 400 data points in the "hull" of an "atom". The density of the points in the core is by magnitudes bigger than in the hull. The inner cluster variance of the hull points is also larger as the distances between the clusters. The data sets can be obtained from the author from http://www.uni-marbug.de/~databionics.

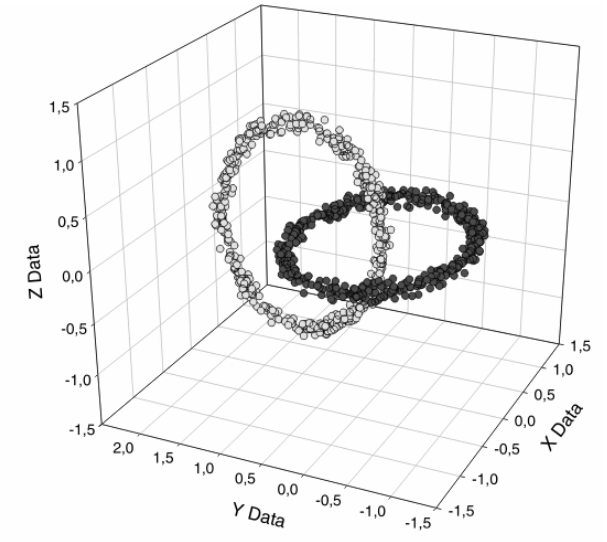

Figure 12: The Chainlink data set

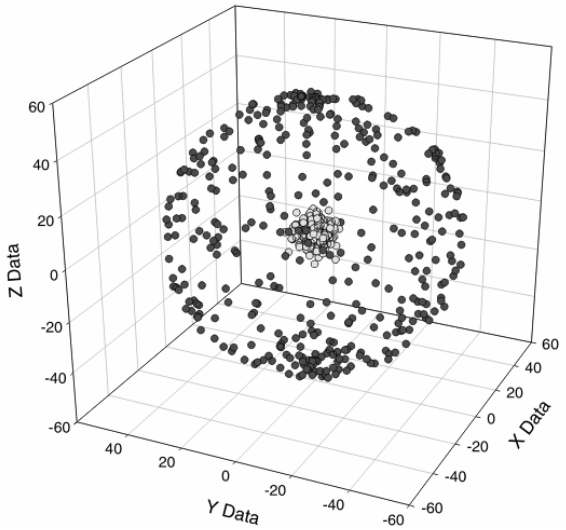

Figure 13: The Atom data set.

7. An efficient movement strategy for effective clustering

Similar to the neighbourhood function used in SOM [9], the field of vision is defined as a circle around a DataBot of radius r on the UD-Matrix. This radius is set to encompass the whole UD-Matrix in the beginning of a simulation. The radius r is successively reduced until only the immediate neighbours of a DataBot are seen. Within it's field of vision a DataBot classifies other DataBots as "friends" or "foes" with respect to the rank of the distances of the pheromones. Other DataBots having pheromones with distances up the so called ParetoPercentile u [7] are termed "friends". The (1-u)-percentile of distances determines the "foes". The "FriendsAndFoes" movement program calculates an attractive force for friends and an repelling force for foes [6]. The forces are proportional to the inverse of the distances d^{-1} of the pheromones. The resulting total force increases the probability of movement in the direction of the force.

The FriendsAndFoes strategy clusters not only Hexa but also data sets of complex cluster structure like the Lsun, TwoDiamonds, Chainlink and the Atom data set. The final positions of the DataBots reached by the FriendsAndFoes strategy and the U-Matrix for Chainlink and Atom can be seen in Figure 14 resp. 15. These pictures are toroid, i.e. the top continues at the bottom and the left side at the right. It can be seen that the clusters are identified correctly by the DataBots. For these data sets the DataBots form a nonlinear mapping from R^3 to R^2. This mapping is cluster-preserving for these data sets. The clusters are nonlinearily disentangled such that clusters can be found in convex regions on the UD-Matrix. The larger inner cluster distances for the "hull" cluster can be seen as nonzero heights of the U-Matrix at the right side of Figure 15. The core of the atom is easily identified in Figure 15.

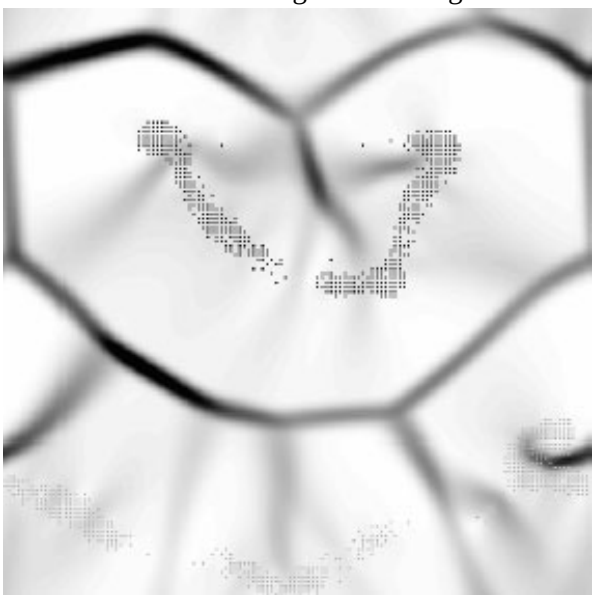

Figure 14: Databots clustering of Chainlink

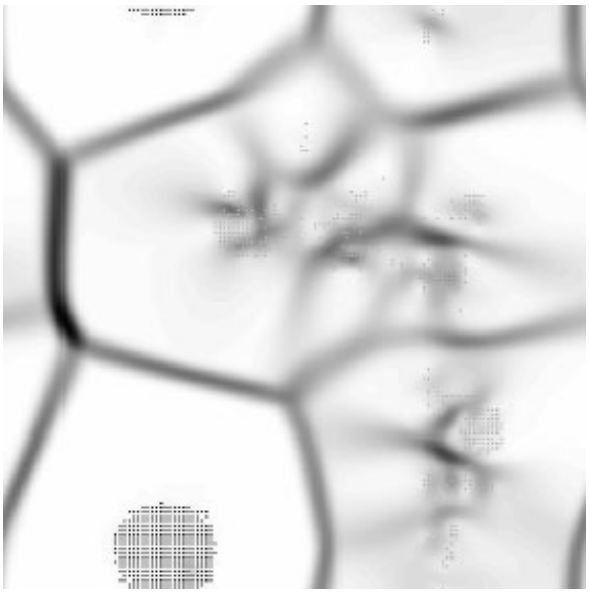

Figure 15: DataBots clustering of Atom

The famous Iris data of Fischer is also correctly clustered using FriendsandFoes [6].

A successful application of the DataBots clustering to data sets of Italian olive oil with other movement programs has been reported in [12]. The FriendsandFoes strategy clusters this data in considerably less simulation steps as any other strategy we tried. The clusters found for the olive oils model the geographic relationships of the origins of the olives.

The clustering ability of the FriendsandFoes strategy can be further enhanced by allowing a bigger portion of random walks at regular intervals during a simulation. These random walks disturb the clustering achieved so far a little bit but allow DataBots captured inside a group to which they do not belong to escape and find their correct cluster. This resembles strongly algorithms that search for a global minimum using simulated annealing.

8. DataBots with FriendsAndFoes strategy applied to stock market analysis

For 1.555 companies with stocks trade on major US-stock markets 9 variables have been collected. These variables consist of fundamental data such as the increase in earnings of the company during the last year and the last three months. Some of the variables also were related to the company's stock price at the beginning of March 2003. The data sets were classified into three classes: if the stock price of a particular company rose by more than 10% during a period of 60 marked days following March 1st 2003 the stock was termed a "winner". If the price dropt more than 5% during that period, the stock was classified a "loser". A stock was classified to be "equal" if it was neither a winner nor a loser. The prior probabilities for the stock in this period were p(winner) = 0.21, p(loser) = 0.6, p(equal) = 0.19. Figure 16 shows a projection of the stocks data classified in this way to three dimensions. It can be seen that clusters in this data set are not easy to find.

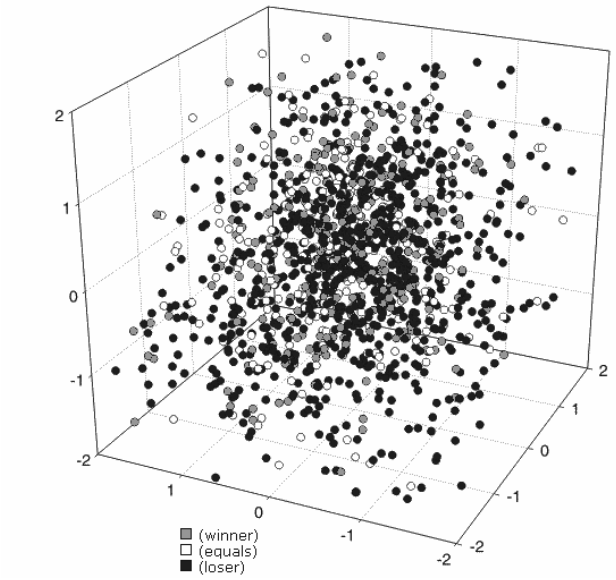

Figure 16: Stocks with winner/loser classification

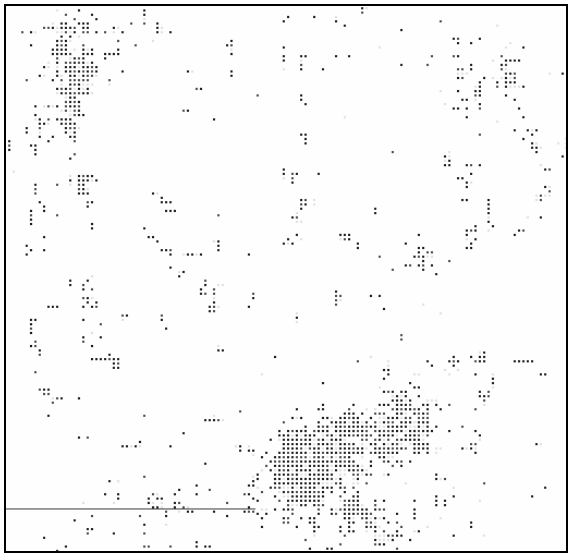

Figure 17: DataBots for Stocks

Figure17 shows the DataBot clustering of the stocks data. Besides several smaller "outliers" two large clusters could be identified. See top left and bottom right in Figure 17. In one of the clusters the number of winners was increased, in the other the number of losers[6]. Both increases were statistically significant for a binomial model on a 5% error level. This means, that it is worthwhile to compare the properties of the cluster in order to find variables that would determine rise or fall of stock prices.

9. Application to cDNA microarray data

The Yeast data published by [13] consists of the expression level of 6153 genes measured in 79 different experiments. From other works, e.g. [14], it is known to be a difficult clustering task. A U-Matrix on the Yeast data set from [14] is shown in Figure 18. The cluster structure of the Yeast data set is rather complicated. This reflects the fact that most of the genes are not altered by the conditions of the cDNA microarray experiments [13]. The expression level of genes that were altered by the experimental conditions follow a highly nonlinear distribution.

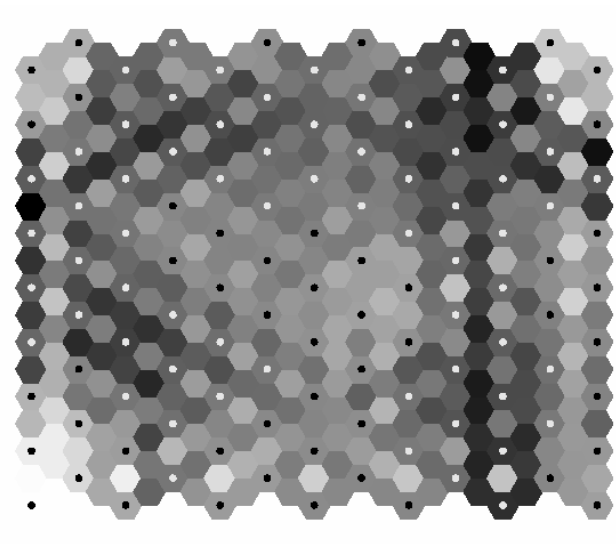

Figure 18 U-Matrix of Yeast data from [14]

Figure 19: DataBot clustering of Yeast

In the densely populated regions of the DataBot clustering of this data set (see Figure 19 left side) the genes not altered by the experiments are shown. The other structures of Figure 19 point to possible new findings on gene expression patterns. Compared to the U-Matrix shown in Figure 18 the DataBots show more structural subtleties of the 79 dimensions of the data set.

10. Conclusion

In [4] a general framework for clustering with DataBots was introduced. In this paper we have exploited two limits of this framework. The MinimalStrategy found outlines the necessary ingredients for successful clustering with swarm systems [6]. The FriendsandFoes strategy has been shown to be efficient on difficult synthetic and real world data. On real data with known clusters, e.g. the Iris data and the data on Italian olive oil, this strategy found the given clusters.

Two real world applications which are known to be difficult to cluster, stock market analysis and cDNA microarray analysis were presented. In both cases the strategy converged to clusters which are interesting for further research. In the case of fundamental stock market analysis the clusters did correlate with expectations about rise and fall of the corresponding stocks.

The efficient and successful movement strategy for DataBots relies on the “cooperation in short range, concurrency in long range”-principle found in many self-organizing systems. The role of random walks for escaping local minima of the ordering could also be confirmed. Theoretical work on information optimal sets a mathematical basis for the definition of short- vs. long range interactions. The consequences of this theory correspond to the so called “Pareto 80/20” law, which is found empirically in many domains.

Future work will include the exploitation of data density in form of the Pareto density [7] and the normalization of the toroid display by fixing the DataBot with a pheromone in the most dense data region to the centre of the display.

Acknowledgement

The DataBot simulations and visualizations were implemented by Dirk Malorny as part of his Diploma Thesis. The visualisations of DataBots and many of the figures were provided by [6]. Figure 1 was designed by Thomas Reiniger.

7. References

[1] E. Bonabeau, M. Dorigo, G. Théraulaz: Cemetery Organization,Brood Sorting, Data Analysis and Graph Partitioning, in Swarm Intelligence: From Natural to Artificial Systems, Santa Fe Institute in the Sciences of the Complexity, Oxford Univ. Press, New York. Transactions on Neural Networks, Vol. 13, No. 1, pp. 3-14, 1999.

[2] E.D. Lumer, B. Faieta , "Diversity and Adaptation in Populations of Clustering Ants". In D. Cliff, P. Husbands, J. Meyer, and S. Wilson (Eds.), Procs. of SAB´94 – 3rd Conf. on Simulation of Adaptive Behavior: From Animal to Animats, Cambridge, MA: The MIT Press/Bradford Books, 1994.

[3] Vitorino Ramos, Ajith Abraham, Evolving a Stigmergic Self-Organized Data-Mining, forthcoming at ISDA-04, 4th Int. Conf. on Intelligent Systems, Design and Applications, August 26-28, Budapest, Hungary, 2004.

Preprint from http://alfa.ist.utl.pt/~cvrm/staff/vramos/ref_50.html

[4] Ultsch, A.: Clustering with DataBots, Technical Report No. 19/99, Philipps Universität, Marburg,Department of Computer Sciences, Juni 1999.

[5] Ultsch, A.: Visualisation and Classification with Artificial Life , in: Proc. Conf. Int. Fed. of Classification Societies ifcs 2000 11-14. July 2000, Namur, Belgium.

[6] Malorny,D.: Data Mining mit Artificial Life Systemen, Diplomarbeit (in German), Data Bionics Research Group, University of Marburg, December 2003.

[7] Ultsch, A.: Maps for the Visualization of high-dimensional Data Spaces, Proc. Workshop on Self organizing Maps, pp 225 - 230, Kyushu, Japan, 2003

[8] Ultsch, A. : Self-organizing Neural Networks for Visualization and Classification, in: O. Opitz, B. Lausen and R. Klar (Eds.): Information and Classification, Berlin, Springer-Verlag, pp. 307-313.

[9] Kohonen, T.: Analysis of a simple self-organizing process, Biological Cybernetics, 43:59-69, 1989.

[10] Ultsch, A.: Justification of Pareto's 80/20 law and precise limits for an ABC-analysis, Technical Report, Nr 30, (in German), Department of Computer Science, University of Marburg, May 2001.

[11] Ultsch, A., Vetter C.: Self-organizing feature maps versus statistical clustering, a benchmark. Research Report No. 9, Dep. of Mathematics, University of Marburg 1994.

[12] Ultsch, A.: Data Mining as an Application for Artificial Life , in Proc. Fifth German Workshop on Artificial Life, pp 191 - 197, Lübeck, 2002

[13] Eisen M.B., Spellmann P., Brown P., Botstein, D.: Cluster analysis and display of genome-wide expression patterns. Proc Natl Aca Sc U S A 95: 14863—14868, 1998..

[14] Kaski S. et al..: Analysis and visualization of gene expression data using self-organizing maps, Neural Networks, Volume 15 , Issue 8, October 2002