ESANN'2008 proceedings, European Symposium on
Artificial Neural Networks - Advances in Computational Intelligence and Learning
Bruges (Belgium), 23-25 April 2008, d-side publi., ISBN 2-930307-08-0.

Explaining Ant-Based Clustering on the basis of Self-Organizing Maps

Lutz Herrmann and Alfred Ultsch

University of Marburg
Dept. of Mathematics and Computer Science
Databionics Research Group

Abstract. Ant-based clustering is a nature-inspired technique whereas stochastic agents perform the task of clustering high-dimensional data. This paper analyzes the popular technique of Lumer/Faieta. It is shown that the Lumer/Faieta approach is strongly related to Kohonen's Self-Organizing Batch Map. A unifying basis is derived in order to assess strengths and weaknesses of both techniques. The behaviour of several popular ant-based clustering techniques is explained.

1 Introduction

Techniques inspired by flocking behaviour of social insects have attracted a lot of attention in numerous research papers due to the ability to exhibit sophisticated self-organization abilities. A particularly interesting field of application is clustering, i.e. the composition of groups of similar objects. The idea behind ant-based clustering is that autonomous stochastic agents, called ants, move data objects on a low-dimensional regular grid such that similar objects are likely to be placed on nearby grid nodes. ones.

Most popular ant-based methods are based on the algorithm proposed by Lumer and Faieta (LF, [4]), whereas ATTA (Adaptive Time Dependent Transporter Ants, [2]) is the most advanced LF derivative. LF-like methods are known for at least two flaws: results are highly dependent on parametrization [1] and even ATTA has found to be "not competitive to the established methods of Multi-dimensional Scaling or Self-Organizing Maps" [2] in terms of topographic mapping.

In sections 2 and 3 the most frequently adapted ant-based clustering algorithm LF is described in a notation consistent with Batch-SOM. A unifying representation for both methods is therefore derived. In section 4 behaviour of ant-based clustering methods is explained using the previously introduced representation.

2 Ant-Based Clustering

The algorithm proposed by Lumer and Faieta (LF, [4]) operates on a fixed regular low-dimensional grid $\mathbb{G} \subset \mathbb{N}^2$. A finite set of input samples X from a vector space with norm $\|.\|$ is projected onto the grid by $m : X \to \mathbb{G}$. The mapping m is altered by autonomous stochastic agents, called ants, that move input samples $x \in X$ from $m(x)$ to new location $m'(x)$. Ants move randomly

ESANN'2008 proceedings, European Symposium on
Artificial Neural Networks - Advances in Computational Intelligence and Learning
Bruges (Belgium), 23-25 April 2008, d-side publi., ISBN 2-930307-08-0.

on neighbouring grid nodes. Ants might pick input samples when facing occupied nodes and drop input samples when facing empty nodes. The probability for picking input sample $x \in X$ from node $i = m(x)$ or dropping picked x on node $j \in \mathbb{G}$ is $p_{pick,x}(i) = \left(\frac{k_1}{k_1+\phi_x(i)}\right)^2$ and $p_{drop,x}(j) = \left(\frac{\phi_x(j)}{k_2+\phi_x(j)}\right)^2$, respectively. Here, $k_1, k_2 \in \mathbb{R}^+$ are threshold constants. $\phi_x(i)$ denotes the average similarity between $x \in X$ and input samples located on the so-called perceptive neighbourhood. Usually, the perceptive neighbourhood consists of $\sigma^2 \in \{9, 25\}$ quadratically arranged nodes at which the ant is located in the center. The set of input samples mapped onto the perceptive neighbourhood around $i \in \mathbb{G}$ is denoted with $N_x(i) = \{y \in X : y \neq x,\ m(y) \text{ neighbouring } i\}$. ϕ is referred to as *local error function* since it represents the local distortion made by $m : X \to \mathbb{G}$. LF-like methods lead to a local sorting of input samples on the grid in terms of similarities. Ants gather scattered input samples into dense clusters (see figure 1 for illustration).

$$\phi_x(i) = \frac{1}{\sigma^2} \sum_{y \in N_x(i)} \left(1 - \frac{\|x-y\|}{\alpha}\right) \tag{1}$$

Popular derivatives of LF are ACLUSTER, ACA and ATTA. ACLUSTER by Ramos et al [5] faces the idle time problem of ants seeking input samples to pick up by introducing simulated pheromones. Ants are more likely to move along pheromone trails where input samples are located. The local error function ϕ remains unchanged. ACA by Vizine et al [10] comes with a cooling scheme for picking probabilities and, therefore, improves convergence. Again, the local error function ϕ remains unchanged.

ATTA (Adaptive Time Dependent Transporter Ants, [2]) has a time-dependent local error function. During an interlude, local error ϕ is altered into $\phi'_x(i) = \frac{1}{|N_x(i)|} \sum_{y \in N_x(i)} (1 - \frac{\|x-y\|}{\alpha})$. To obtain a better topographic mapping, normalization is done with $|N_x(i)|$ instead of σ^2. The effect of the altered local error function is discussed in section 4.

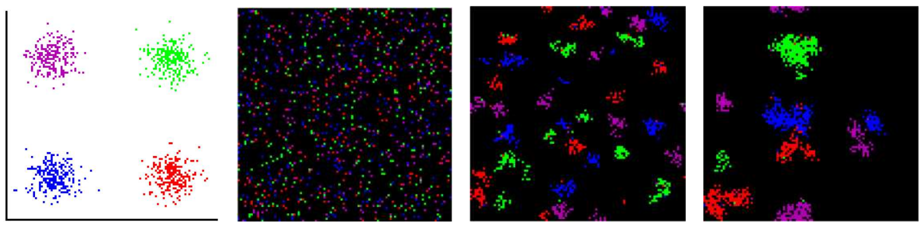

Fig. 1: typical results [1] of LF algorithm from left to right: gaussian data with 4 clusters, initial mapping of data objects, dense clusters appear, too many clusters with topological defects have emerged

3 Ant-Based Clustering vs. Self-Organizing Maps

Self-Organizing Batch Maps (Batch-SOM, [3]) are well-known artificial neural networks that consist of a grid $\mathbb{G}$, codebook vectors $w_i \in \mathbb{R}^n, i \in \mathbb{G}$ and a mapping function $m : X \to \mathbb{G}$ with $m(x) = \arg\min_{i \in \mathbb{G}} \|x - w_i\|$. The codebook vectors are defined according to equation 2 at which $h : \mathbb{G} \times \mathbb{G} \to [0, 1]$ denotes a time-depending neighbourhood function. An update of $m : X \to \mathbb{G}$ leads to an update of codebook vectors $w_i, i \in \mathbb{G}$ and vice versa. This is how the Batch-SOM modifies mapping $m : X \to \mathbb{G}$. For details see [3].

$$w_i = \frac{\sum_{x \in X} h(m(x), i) \cdot x}{\sum_{x \in X} h_{m(x)}(i)} \tag{2}$$

In order to compare Batch-SOM and LF, a unifying basis for both algorithms is derived. Input data X and output space $\mathbb{G}$ are identical and mapping function $m : X \to \mathbb{G}$ is iteratively update in both cases as well.
A local error for the Batch-SOM is derived from the quantization error $\|x - w_i\|$ because its minimization determines the update of $m : X \to \mathbb{G}$. Resolving the quantization error with equation 2 leads to the local error function Φ of the Batch-SOM (see equation 3). Φ_x represents the norm of averaged differences $x - y$ of grid-neighbouring input samples $y \in X$.

$$\Phi_x(i) = \frac{\left\| \sum_{y \in X} h(m(y), i) \cdot (x - y) \right\|}{\sum_{y \in X} h(m(y), i)} \tag{3}$$

In the following, the mechanism of picking and dropping ants is no longer subject of consideration. In [6] it was shown that collective intelligence can be discarded in LF systems, i.e. same results could be achieved without ants but using the local error ϕ directly for probabilistic cluster assignments. This simplification is evident: over a period of time, randomly moving ants may select an arbitrary subset of input samples but re-allocation through picking and dropping depends on ϕ only. Probability of selection is the same on all input samples such that ants might be omitted in favor of any other subset sampling technique.
A meaningful symmetrical neighbourhood function $h : \mathbb{G} \times \mathbb{G} \to [0, 1]$ for the LF algorithm is defined according to the perceptive neighbourhood of ants, i.e. $h(i, j)$ is 1 if $j \in \mathbb{G}$ is located in the perceptive neighbourhood of node $i \in \mathbb{G}$ and 0 elsewhere. This neighbourhood function allows to restate the LF error function ϕ as equation 4 by utilizing $|N_x(i)| = \sum_{y \in X} h(m(y), i)$.

$$\phi_x(i) = \frac{|N_x(i)|}{\sigma^2} \cdot \left(1 - \frac{\Phi'_x(i)}{\alpha}\right) \quad \text{with} \quad \Phi'_x(i) = \frac{\sum_{y \in X} h(m(y), i) \cdot \|x - y\|}{\sum_{y \in X} h(m(y), i)} \tag{4}$$

The local error $\phi = \frac{|N|}{\sigma^2}(1 - \frac{\Phi'}{\alpha})$ of the LF algorithm incorporates a topographic term Φ' that acts as an upper limit to Φ, since $\Phi_x(i) \leq \Phi'_x(i) \forall x, i$. The elimination of codebook vectors, introduction of a SOM-specific local error function

ESANN'2008 proceedings, European Symposium on
Artificial Neural Networks - Advances in Computational Intelligence and Learning
Bruges (Belgium), 23-25 April 2008, d-side publi., ISBN 2-930307-08-0.

and definition of a meaningful neighbourhood function for the LF algorithm lead to the following insights: The LF is a non-deterministic derivative of the Batch-SOM with comparable local error function. For an overview of differences of both methods see table 1.

	Batch-SOM	LF
neighbourhood $h : \mathbb{G} \times \mathbb{G} \to [0, 1]$	large, shrinking	small, fixed
update of $m : X \to \mathbb{G}$	deterministic	probabilistic
local error function	Φ	$\frac{\lvert N \rvert}{\sigma^2}(1 - \frac{\Phi'}{\alpha})$
termination	cooling scheme	never

Table 1: differences of Batch-SOM and LF-algorithm

4 Assessment of Ant-Based Methods

There are three main reasons why ant-based clustering methods following the LF scheme are prone to produce bad topographic mappings, e.g. too many, too small and topographically distorted clusters:
(1) The local error function ϕ consists of two terms. Maximization of the *topographic term* $1 - \frac{\Phi'}{\alpha}$ corresponds to minimization of Φ' and Φ, respectively. This is known to produce sufficiently topography preserving mappings $m : X \to \mathbb{G}$, e.g. in the Batch-SOM [3] [8]. The *output density term* $\frac{|N|}{\sigma^2}$ is not related to the configuration of the available clusters but to the density of the mapped input samples $m(X)$. In comparison to the topographic term, the output density term is much easier to maximize and, therefore, will dominate the local error function ϕ. Obviously, the output density term distorts the correct topographic term.
(2) The topographic term $1 - \frac{\Phi'}{\alpha}$ of the LF local error depends on the shape of the neighbourhood function $h : \mathbb{G} \times \mathbb{G} \to \{0, 1\}$ (see section 3). Usually, the neighbourhoods' sizes are chosen as $\sigma^2 \in \{9, 25\}$, i.e. the immediate neighbours. It is known from the Self-Organizing Maps that too small neighbourhoods prevent a sufficient preservation of topography, i.e. too many and too small clusters emerge during the training process.
(3) Range and variance of the topographic term $1 - \frac{\Phi'}{\alpha}$ both depend on $\alpha \in \mathbb{R}^+$. Obviously, α is a tradeoff parameter that controls how to weight output space density and topographic projection quality. Usually this is a unknown and data-dependent quality, and, therefore, the LF method should be avoided for topographic mappings.

The ACLUSTER algorithm [5] and ACA [10] have both shown to produce too many, too small and topographically distorted clusters. Ramos et al. [5] [1] noticed that the LF algorithm generates a large quantity of small clusters. The applied local error function ϕ explains this behaviour: the accountancy of output space densities and too small perceptive neighbourhoods both distort the iterative optimization of topographic mappings.

Aranha and Iba [1] have faced the problem of parametrization by using a genetic algorithm to optimize several parameters of the LF algorithm, e.g. $k_1, k_2, \alpha \in \mathbb{R}$. Some minor improvements against the parameters found in literature were claimed to be made. Since the problem of output density distorted ϕ was not tackled, illustrative examples still show topographically distorted mappings (see figure 1).
In contrast to ordinary LF methods, the ATTA algorithm [2] uses a time dependent local error function (see section 2). During an interlude the local error function $\phi = \frac{|N|}{\sigma^2}(1 - \frac{\Phi'}{\alpha})$ is altered into $\phi' = 1 - \frac{\Phi'}{\alpha}$ which is similar to the function of the Batch-SOM. During that interlude, formed clusters disperse and the assembly of input samples changes into a more topography preserving configuration (see figure 2 for illustration) at which positionings of input samples strongly resemble bestmatch coordinates on Emergent Self-Organizing Maps (see [8]). Nevertheless, compared to Self-Organizing Maps the ATTA algorithm produces poor results in terms of topographic mappings during experiments carried out on both synthetic and real data [2]. The authors stated "ant-based clustering and sorting not to be a satisfactory method for topographic mapping".

5 Discussion

This work shows a previously unknown relation of two topographic mapping techniques. The assessment of algorithms conforms with experimental results from other researchers. Yet, it is based on the assumption [6] that stochastic agents, e.g. ants, are nothing more than an arbitrary sampling technique that is to be omitted for further analysis of formulae. This simplification is evident but may be invalid for ants guided by more than just randomness and ϕ , e.g. pheromone trails. Our analysis of formulae does not cover popular algorithms that are not LF-derivatives. DataBots [7] operate on a weighted sum of rank-based local error functions, whereas Cellular Ants [9] behave similar to Schelling model automats, according to discretized similarities.

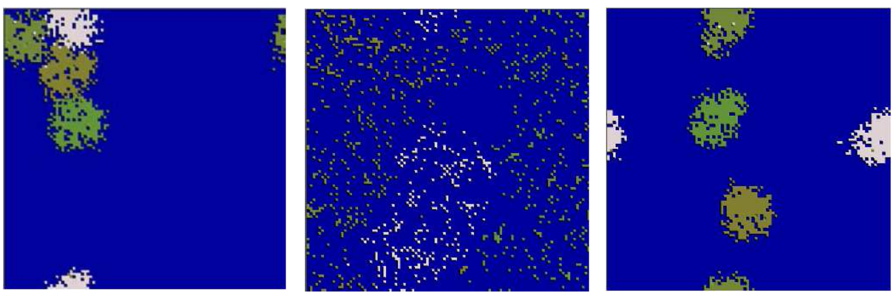

Fig. 2: ATTA algorithm [2], from left to right: dense clusters have emerged; altered local error ϕ' leads to ESOM-like mapping during interlude; separated dense clusters appear after interlude because of classical local error function ϕ

ESANN'2008 proceedings, European Symposium on
Artificial Neural Networks - Advances in Computational Intelligence and Learning
Bruges (Belgium), 23-25 April 2008, d-side publi., ISBN 2-930307-08-0.

6 Summary

Up to our knowledge, this is the first work that shows how the ant-based clustering algorithm from Lumer and Faieta (LF, [4]) is related to Self-Organizing Maps [3]. We omit the mechanism of picking and dropping ants for a formal analysis of the underlying formulae and compare it with Kohonen's well-known Batch-Map. In order to denote both algorithms on a unifying basis, local error functions were derived for both methods. The local errors of LF algorithms are closely related to the one of the Batch-SOM. Therefore, we consider the LF algorithm to be a probabilistic derivative of the Batch-SOM.
Furthermore, it is possible to explain the behaviour of LF and derivatives on that basis. Our prediction of results agrees with the experimental results achived by other researchers. The original LF algorithm is prone to produce too many, too small and topographically incorrect clusters compared to the SOM. The ATTA algorithm is a hybrid showing SOM-like behaviour because of its SOM-like local error function. LF and derivatives can easily be improved using altered neighbourhood functions and normalization schemes.

References

[1] C. Aranha, H. Iba, The effect of using evolutionary algorithms on ant clustering techniques, In The Long Pham and Hai Khoi Le and Xuan Hoai Nguyen (editors), *Proceedings of the Third Asian-Pacific workshop on Genetic Programming*, pages 24-34, Military Technical Academy, Hanoi, VietNam, 2006.

[2] J. Handl, J. Knowles, M. Dorigo, Ant-Based Clustering and Topographic Mapping, *Artificial Life 12(1)*, MIT Press, Cambridge, MA, USA, 2006.

[3] T. Kohonen, Self-Organizing Maps, Springer Series in Information Sciences, Vol. 30, Springer, Berlin, Heidelberg, New York, 1995, 1997, 2001,

[4] E. Lumer, B. Faieta, Diversity and adaption in populations of clustering ants, In Proceedings of the *Third International Conference on Simulation of Adaptive Behaviour: From Animals to Animats 3*, pages 501-508, MIT Press, Cambridge, MA, 1994.

[5] V. Ramos, F. Muge, P. Pina. Self-organized data and image retrieval as a consequence of inter-dynamic synergistic relationships in artificial ant colonies. In: Ajith Abraham, Javier Ruiz-del Solar and Mario Koppen (Eds.), *2nd Int. Conf. on Hybrid Intelligent Systems*, Vol. 87, pages 500-509, Santiago, Chile, IOS Press, 2002.

[6] S. C. Tan, K. M. Ting, S. W. Teng, Reproducing the Results of Ant-Based Clustering Without Using Ants, *IEEE Congress on Evolutionary Computation*, 2006.

[7] A. Ultsch, *Clustering with DataBots*, Technical Report No. 19, Dept. of Mathematics and Computer Science, University of Marburg, Germany, 1999.

[8] A. Ultsch, L. Herrmann, The architecture of emergent self-organizing maps to reduce projection errors, In: Verleysen M. (Eds), *Proceedings of the European Symposium on Artificial Neural Networks (ESANN 2005)*, 2005.

[9] A. Vande Moere, J. J. Clayden, Cellular Ants: Combining Ant-Based Clustering with Cellular Automata, In: *17th IEEE International Conference on Tools with Artificial Intelligence, ICTAI'05*, 2005.

[10] A. L. Vizine, L. N. de Castro, R. R. Gudwin, Text Document Classification Using Swarm Intelligence, In: *International Conference on Integration of Knowledge Intensive Multi-Agent Systems*, pages 134 – 139, 2005.