The Architecture of Ant-Based Clustering to improve Topographic Mapping

Lutz Herrmann and Alfred Ultsch

Databionics Research Group, Dept. of Mathematics and Computer Science
University of Marburg
{lherrmann, ultsch}@informatik.uni-marburg.de

Abstract. This paper analyzes the popular ant-based clustering approach of Lumer/Faieta. Analysis of formulae unveils that ant-based clustering is strongly related to Kohonen's Self-Organizing Batch Map. Known phenomena, e.g. formation of too many and too small clusters, can be explained due to that. Furthermore it is shown how topographic mapping of ant-based methods is substantially improved by means of a modified error function. This is demonstrated on few selected fundamental clustering problems.

1 Introduction

Techniques inspired by flocking behaviour of social insects have attracted a lot of attention in numerous research papers over the last decade due to the ability of simple interacting entities to exhibit sophisticated self-organization abilities. 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 more likely to be placed on nearby grid nodes than dissimilar ones.

Most popular methods are based on the algorithm proposed by Lumer and Faieta [6]. Lumer/Faieta derivatives are known for at least two flaws: results highly depend on parametrization [1] and have been found to be "not competitive to the established methods of Multi-dimensional Scaling or Self-Organizing Maps" [3] in terms of topographic mapping.

This paper shows how to analyze ant-based clustering methods on basis of Self-Organizing Maps. A unifying representation for Lumer/Faieta and well-known Self-Organizing Batch Maps is introduced. Naive improvements for topographic mappings of ant-based methods are derived and experimentally verified.

2 Ant-Based Clustering

The method proposed by Lumer and Faieta [6] (here after LF algorithm) 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 might pick input samples when facing occupied nodes and drop input samples when facing empty nodes. Probabilities for picking and dropping are computed using the average similarity $\phi_x(i)$ between $x \in X$ and input samples located on the so-called perceptive neighbourhood around node $i \in \mathbb{G}$. The perceptive neighbourhood usually 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 $N_x(i) = \{y \in X : y \neq x,\ m(y) \text{ neighbouring } i\}$. In this context, ϕ is referred to as *error function* since its minimization determines the ants' probabilistic modifications of mapping m.

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

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 crowds. In literature, it has been noticed that LF and its derivatives are prone to produce too many and too small clusters [1] [3]. For illustration see Figure 1.

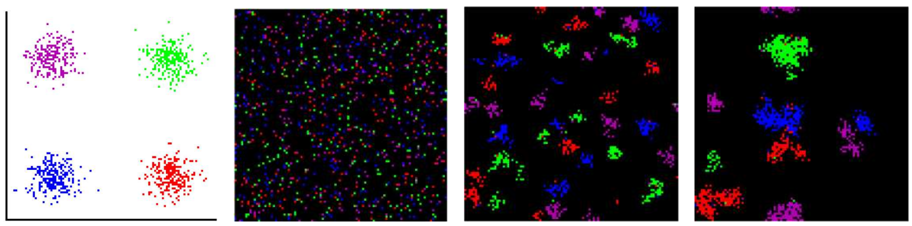


Fig. 1. Typical results [1] of LF algorithm from left to right: gaussian data with 4 clusters, initial mapping of data objects, dense clusters appear, finally too many clusters with topological defects have emerged.

3 Analysis of Ant-Based Clustering by means of Self-Organizing Batch Maps

Self-Organizing Batch Maps (Batch-SOM, [5]) are well-known artificial neural networks that consist of 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-dependent 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 [5].
In literature, two main types of Self-Organizing Maps (SOM) can be distinguished: first, SOM in which each codebook vector represents a single cluster.

In contrast to that, SOM consisting of several thousands of codebook vectors visualize structural features of the input space These SOM are referred to as Emergent Self-Organizing Maps (ESOM, [9]).

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

A meaningful error function 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 error function Φ of the Batch-SOM. 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 [7] it was shown that collective intelligence can be discarded in LF systems, i.e. same results could be achieved without ants but using error function 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.

For the LF algorithm a meaningful neighbourhood function $h : \mathbb{G} \times \mathbb{G} \to [0, 1]$ 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 ϕ as follows:

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

The error function $\phi = \frac{|N|}{\sigma^2}(1 - \frac{\Phi'}{\alpha})$ of the LF algorithm incorporates the term Φ' that is a weighted sum of local input space distances. Obviously, Φ' measures the local stress of topographic mapping $m : X \to \mathbb{G}$, comparable to the error function Φ of the Batch-SOM. Φ' even acts as an upper limit to Φ since $\forall x \in X, i \in \mathbb{G} : \Phi_x(i) \leq \Phi'_x(i)$. Due to that Φ' is referred to as *topographic term* of the LF algorithm. The term $\frac{|N_x(i)|}{\sigma^2}$ estimates the output space density around grid node $i \in \mathbb{G}$. Therefore, it is referred to as *output density term* of the LF algorithm.

A unifying framework for analysis and assessment of Batch-SOM and LF exists by means of error functions Φ and ϕ. Both error functions are denoted by means of three functions: norm $\|.\|$, neighbourhood $h : \mathbb{G} \times \mathbb{G} \to [0, 1]$, and mapping $m : X \to \mathbb{G}$. This leads to the following insights: The LF algorithm uses a fixed neighbourhood function with small radius, whereas Batch-SOM

Table 1. Differences of Batch-SOM, LF-algorithm.

	Batch-SOM	Lumer/Faieta
neighbourhood $h : \mathbb{G} \times \mathbb{G} \to [0, 1]$	large, shrinking	small, fixed
update of $m : X \to \mathbb{G}$	deterministic	probabilistic
searching for update of $m : X \to \mathbb{G}$	global $\mathbb{G}$	local $N_{m(x)} \subset \mathbb{G}$
error function	Φ	$\frac{\vert N\vert}{\sigma^2}(1 - \frac{\Phi'}{\alpha})$
termination	cooling scheme	never

uses shrinking neighbourhood functions with large radiuses. The LF algorithm has a probabilistic update of mapping $m : X \to \mathbb{G}$, whereas Batch-SOM is deterministic. The error function of the LF algorithm decomposes into an output density term $\frac{|N|}{\sigma^2}$ and a topographic term $1 - \frac{\Phi'}{\alpha}$. The topographic term is easily identified as a topographic distortion measure because of its relation to Batch-SOM error Φ. Therefore, the LF algorithm is easily convertible into a special case of Batch-SOM, and vice versa. For an overview of differences see Table 1.

4 Assessment and Improvement

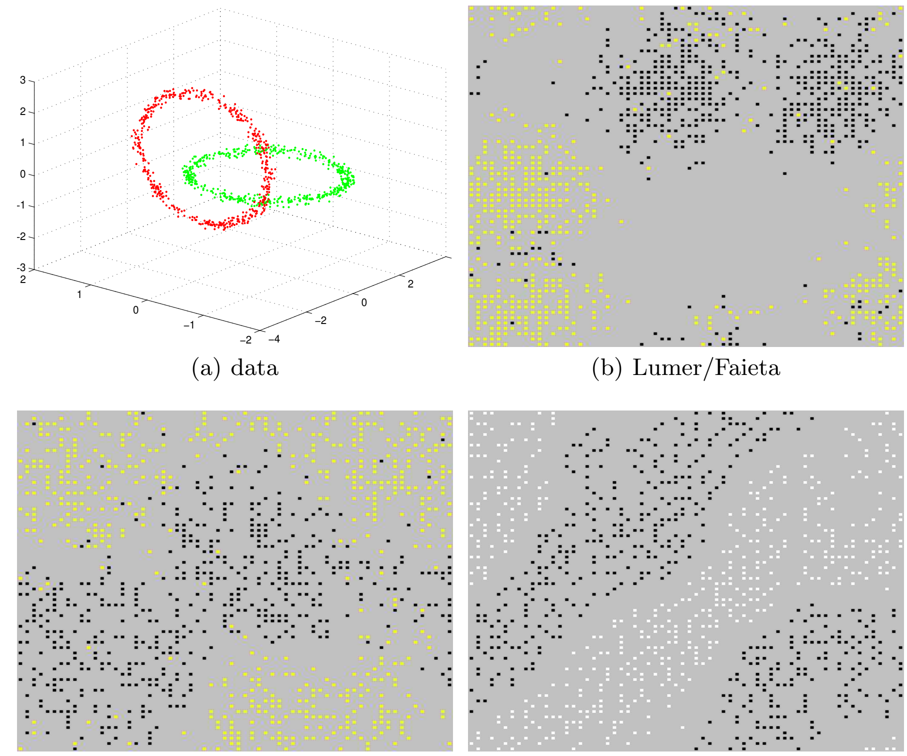
Ant-based clustering methods following the LF scheme are prone to produce bad topographic mappings, e.g. too many, too small and topographically distorted clusters. If one regards LF as a derivative of the Batch-SOM, improvement of topographic mapping can easily be achieved.

Maximization of the *topographic term* $1 - \frac{\Phi'}{\alpha}$ corresponds to minimization of Φ' and Φ, too. This is known to produce sufficiently topography preserving mappings $m : X \to \mathbb{G}$, e.g. when using Batch-SOM [5].

In contrast to that, the *output density term* $\frac{|N|}{\sigma^2}$ has two mayor flaws. First, maximization of output space densities does not imply any preservation. Obtained mappings are, therefore, not related to the configuration of available clusters in the input space. In additon to that, the LF algorithm is not allowed to assign two or more objects to a single grid node (see Section 2) in order to prevent the mapped clusters from collapsing into a single grid node. Due to that, densities of input data are hardly preservable on grid $\mathbb{G}$.

In comparison with the topographic term, the output density term is much easier to maximize and, therefore, will distort the error ϕ. Accounting of output densities is prone to distort the formation of correct topographic mappings because it is responsible for additional local optima of ϕ. Future derivatives of the LF algorithm should maximize $1 - \frac{\Phi'}{\alpha}$ and minimize Φ', respectively, in order to obtain better topographic mappings. For example, Figure 2 illustrates that traditional LF does not preserve looped cluster structures, in contrast to Emergent SOM and modified LF without density term.

The topographic term $1 - \frac{\Phi'}{\alpha}$ of the LF error depends on the shape of the neighbourhood function $h : \mathbb{G} \times \mathbb{G} \to [0, 1]$ (see Section 3). Usually, the neighbour-



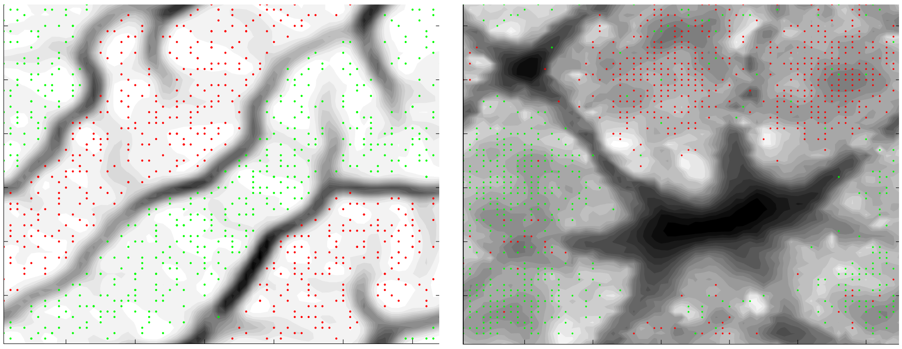
(a) data (b) Lumer/Faieta

(c) LF, no density term (d) Emergent SOM

Fig. 2. Looped chainlink data from FCPS [8], data mapped on grid by several methods, only Emergent SOM and "LF without density term" enable formation of looped clusters with little effort.

hoods' sizes are chosen as $\sigma^2 \in \{9, 25\}$, i.e. the immediate neighbours. From the Self-Organizing Batch Map (Batch-SOM) it is known that the cooling scheme of the neighborhood radius influences the goodness for topographic mapping very strongly (see [4] for details). A bigger radius enables a more continuous mapping in the sense that proximities existing in the original data are visible on the grid. This is evident because smaller neighbourhoods are more likely to exclude parts of a cluster.

In order to prevent future LF derivatives from unsufficient topographic mappings, i.e. too many and too small clusters emerge during the training process, bigger neighbourhood radiuses are to be chosen. The ideal learning radius, however, remains a data-dependent quality.



(a) Emergent SOM (b) Lumer/Faieta

Fig. 3. U-Maps of looped chainlink data, input space distances depicted as gray levels (a) looped clusters, large distances occur between clusters (b) no looped clusters, large distances occur even inside clusters due to accounting for output density.

5 Advanced Topographic Mapping

LF derivatives that do not account for output densities, usually will produce a SOM-like, equally distributed mapping of input samples (see Figure 2 for illustration). In this case, cluster retrieval cannot be achieved according to sparse regions dividing dense clusters on the grid.

A promising technique for cluster retrieval is based on so-called U-Maps [9]. Arbitrary projections from normed vector spaces onto grid $\mathbb{G}$ are transformed into landscapes, so-called U-Maps. The U-Map technique assigns each grid node a height value that represents the averaged input space distance to its' neighbouring nodes and codebook vectors, respectively. Clusters lead to valleys on U-Maps whereas empty input space regions lead to mountains dividing the cluster valleys (see Figure 3 for illustration). The U*C cluster algorithm uses the so-called watershed transformation to retrieve cluster valleys on U-Maps (see [10] for details).

6 Experimental Settings and Results

In order to measure the distortion of a given topographic mapping method, a collection of fundamental clustering problems (FCPS) is used [8]. Each data set represents a certain problem that arbitrary algorithms shall be able to handle when facing unknown real-world data. Here, two versions of the Lumer/Faieta approach are tested on which one delivers the best topographic mapping: with and without accounting for output density.

A comprehensive overview on topographic distortion measurements can be found in [2]. Here, the so-called *minimal path length* (MPL) measurement is used. It is an easy-to-compute measurement that sums up input space distances of grid-neighbouring data objects and codebook vectors, respectively.

$$mpl = \sum_{x \in X} \frac{1}{|N_x|} \sum_{y \in N_x} \|x - y\| \tag{5}$$

Lower MPL values indicate less topographic distortion when moving on the grid and, therefore, a more trustworthy topographic mapping. Each algorithm is run several times with the same parametrization. MLP values indicate if accounting for output densities assists the formation of good topographic mappings, or not. All data sets from the FCPS collection were processed with the same parameters established in literature, i.e. $\alpha = 0.5$, $\sigma^2 = 25$, $k_1 = 0.3$ and $k_2 = 0.1$ on a 64×64 grid with 100 ants during 100000 iterations. The results can be found in Table 2. Accounting for output densities leads to increasing MPL values on an average, i.e. worsenings of topographic mappings. Significance has been confirmed using a Kolmogorov-Smirnov test on a $\alpha = 5\%$ level.

Table 2. Topographic distortion measured by *minimal path length* method, mean values ± standard deviation of each 100 experiments, p-values of Kolmogorov-Smirnov test indicate that "LF without density term" produces significantly smaller error values than traditional LF.

data set	LF with density term		LF without density term	p-value
atom	161 ± 15.6	$>$	142 ± 6.2	$1.24E{-}12$
chainlink	6.33 ± 0.33	$>$	6.19 ± 0.12	$1.38E{-}05$
hepta	11.16 ± 0.66	$>$	9.86 ± 0.54	$2.65E{-}13$
iris	11.8 ± 0.65	$>$	10.02 ± 0.57	$1.03E{-}17$
target	6.69 ± 0.41	$>$	5.35 ± 0.33	$8.79E{-}23$
2diamonds	3.86 ± 0.09	$>$	3.28 ± 0.10	$1.08E{-}23$
wingnut	5.64 ± 0.32	$>$	5.07 ± 0.23	$9.91E{-}11$

7 Discussion

Minimal path lengths (MPL), as proposed in Section 6, are well-known topographic distortion measures. The length of *paths* is normalized by the cardinality $|N_x|$ of the corresponding grid neighbourhood, i.e. the number of objects mapped onto the grid neighbourhood. This is supposed to decrease error values of locally dense mappings, as produced by traditional LF, because small radial neighbourhoods usually do not cover objects of another cluster, since locally dense mappings imply sparse dividing grid regions around clusters. Nevertheless, traditional LF produces bigger MPL errors than the modified LF that is not accounting for densities. We conclude that the topographic mapping quality is improved beyond our empirical evaluation.

8 Summary

To the best of our knowlege, this is the first work that shows how the LF algorithm by Lumer and Faieta [6] is related to Self-Organizing Maps [5]. The mechanism of picking and dropping ants was omitted in favor of a formal analysis of the underlying formulae and comparison with Kohonen's Batch-SOM. It could be shown that a unifying framwork for both methods does exist in terms of closely related topographic error functions. The LF algorithm is to be considered a probabilistic, first-class relative of the Batch-SOM. The behaviour of LF and derivatives becomes explainable on that unifying basis.

Ant-based clustering methods derived from LF exhibit poor clustering abilities because of distorted topographic mappings. Improvements of topographic mapping were derived by means of SOM architecture. Perceptive areas are to be increased, and accounting for density of mapped data is futile. The obtainable methods do not produce dense clusters any more but equally distributed, SOM-like mappings. Due to that, cluster are to be retrieved using U-Map technology. As predicted by our theory, an empirical evaluation showed on few clustering problems that not-accounting for density of mapped data improves the quality of topographic mapping despite of unfavorable settings.

References

1. Aranha, C., Iba, H.: The effect of using evolutionary algorithms on ant clustering techniques. Proc. Third Asian-Pacific workshop on Genetic Programming (2006) 24–34
2. Goodhill, G. J., Sejnowski, T. J., Quantifying neighbourhood preservation in topographic mappings. Proc. 3rd Joint Symposium on Neural Computation (1996) 61–82
3. Handl, J, Knowles, J., Dorigo, M.: Ant-Based Clustering and Topographic Mapping. Artificial Life **12** (2006) 35–61
4. Nybo, K., Venna, J., Kaski, S.: The self-organizing map as a visual neighbor retrieval method. Proc. of the Sixth Int. Workshop on Self-Organizing Maps (2007)
5. Kohonen, T.: Self-Organizing Maps. Springer Series in Information Sciences (2001)
6. Lumer, E., Faieta, B.: Diversity and adaption in populations of clustering ants. Proc. Third International Conference on Simulation of Adaptive Behaviour: From Animals to Animats **3** (1994) 501–508
7. Tan, S. C., Ting, K. M., Teng, S. W.: Reproducing the Results of Ant-Based Clustering Without Using Ants. Proc. IEEE Congress on Evolutionary Computation (2006) 1760–1767
8. Fundamental Clustering Problem Suite. `http://www.uni-marburg.de/fb12/datenbionik/data`
9. Ultsch, A., Mörchen, F.: U-maps: topograpic visualization techniques for projections of high dimensional data, Proc. 29th Annual Conference of the German Classification Society (2006)
10. Ultsch, A., Herrmann, L.: Automatic Clustering with U*C. Technical Report, Philipps-University of Marburg (2006)