

7 Behavior-based Systems in Data Science

Many technological advances have been achieved with the help of bionics, which is defined as the application of biological methods and systems found in nature. A related, rarely discussed subfield of information technology is called databionics. *Databionics* refers to the attempt to adopt information processing techniques from nature. This chapter will discuss the imitation of natural processes (also called biomimicry [Benyus, 2002]) using swarm intelligence, which is a form of artificial intelligence (AI) [Bonabeau et al., 1999] and was introduced as a term in the context of robotics [Beni/Wang, 1989]. In the context considered here, AI may be described as a field of study that seeks to explain and emulate intelligent behavior in the form of a computational process³⁴ [Russell et al., 2003, p. 5].

Consequently, *swarm intelligence* is defined as the emergent collective behavior³⁵ of simple entities called agents³⁶ [Bonabeau et al., 1999, p. 12]. An agent is a software entity, situated³⁷ in a given environment, that is capable of flexible, autonomous action in order to meet its design objectives [Jennings et al., 1998]. In the context of swarms, the terms behavior and intelligence are used synonymously, bearing in mind that in general, the definition of intelligence is controversial [Legg/Hutter, 2007] and complex [Zhong, 2010]. The properties of swarm behavior will be explained later in this section.

“There are [...] three key concepts [...] [related to agents]: situatedness, autonomy, and flexibility. Situatedness, in this context, means that the agent receives sensory input from its environment and that it can perform actions which change the environment in some way” [Jennings et al., 1998, p.8].

Autonomy refers to an agent’s capability for independent, decentralized action, and flexibility refers to its ability to proactively respond to its environment in a “timely fashion” [Jennings et al., 1998].

Inspired by Beni’s definition of intelligent robots [Beni/Wang, 1993, p. 705], here, an intelligent agent is described as one whose behavior is neither random nor predictable [Beni, 2004, p. 4]. On the one hand, “intelligent behavior is the production of something ordered, i.e., unlikely to occur: an improbable outcome” [Beni, 2004, p. 3]. On the other hand, unpredictability is not equivalent to intelligence; a roulette, for example, is not intelligent [Beni, 2004, p. 3]. “It seems that somehow both unpredictability and the creation of some order are necessary to be able to speak of “intelligence” [Beni, 2004, p. 3]. In the context of data science, the first intelligent agents to be developed and applied were called DataBots [Ultsch, 2000a]. DataBots possess probabilistically defined movement strategies, take in food, consume food and store quantities of food. However, the question of whether DataBots themselves exhibit swarm intelligence is controversial [de Buitléir et al., 2012, p. 2], and as such, they will be separately introduced in the next section. It will be shown that in the case of swarm-organized projection (SOP)

³⁴ The author focuses on AI in the context of behavior; however, thought process and reasoning types of AI also exist, of which neural networks and Bayesian learning are representative examples.

³⁵ The term collective behavior generically denotes any behavior of agents in a system of more than one agent [Cao et al., 1997].

³⁶ See also a similar definition in [Martens et al., 2011, p. 2].

³⁷ “The word “situated,” [...] is intended to emphasize that the process of deliberation takes place in an agent that is directly connected to an environment” [Russell et al., 2003, p. 422].

[Herrmann, 2011], DataBots do not exhibit swarm intelligence, whereas in the case of Pswarm (introduced in the next chapter), they do.

Another example of the use of intelligent agents is Schelling’s segregation model [Schelling, 1969, 1971]. The model consists of a lattice of square patches (tiling). Agents are located on this landscape, initially at random, with no more than one on any patch. The agents are of two different types, e.g., blue and red, and there are free patches available. Each agent has a tolerance parameter. A blue agent is “happy” when the ratio of blues to reds in its Moore neighborhood (the eight immediately adjacent patches) is above its tolerance threshold. Unhappy agents are allowed to move randomly to a new open position (white). Schelling’s segregation model leads to segregation of the agents, even when individual agents have only a mild preference for living near agents of the same type. An example of the segregation process is illustrated in Figure 7.1.

“Originally the model was intended to explain how racialized city ghettos might emerge from individual choices, given even slight racial biases. Some important constraints on effective segregation have been described by [Vinković/Kirman, 2006]. Segregation is greatly increased if agents are allowed to jump to any node that yields less stress, instead of neighbouring nodes only” [Herrmann, 2011, pp. 54-55].

Swarm behavior can be imitated based on observations of herds [Wong et al., 2014], bird flocks and fish schools [Reynolds, 1987], bats [Yang/He, 2013], or insects such as bees [Karaboga, 2005; Karaboga/Akay, 2009], ants [Deneubourg et al., 1991], fireflies [Yang, 2009], cockroaches [Havens et al., 2008], midges [Passino, 2013], glow-worms or slime moulds [Parpinelli/Lopes, 2011]. [Grosan et al.] define five main principles of swarm behavior: *Homogeneity*, meaning that every agent has the same behavior model; *Locality*, meaning that the motion of each agent is influenced only by its nearest neighbors; *Velocity Matching*, meaning that each agent attempts to match the velocity of nearby flockmates; *Collision Avoidance*, meaning that each agent avoids collisions with nearby agents; and *Flock Centering*, meaning that agents attempt to stay close to neighboring agents [Grosan et al., 2006, p. 2; Reynolds, 1987, pp. 6, 7]. Here, these definitions are given greater specificity in two respects.

First, the term *agent* is modified to the term *agents of the same type* because many swarms consists of more than one type of agent, e.g., small and large workers in the Pheidole genus of ants [Bonabeau et al., 1999, pp. 3, 4]. Second, a swarm need not necessarily move. For example, fire ants self-assemble into waterproof rafts to survive floods [Mlot et al., 2011]. The individual ants are linked together to construct such self-assemblages [Mlot et al., 2011]. Therefore, velocity matching can result in a velocity of zero.

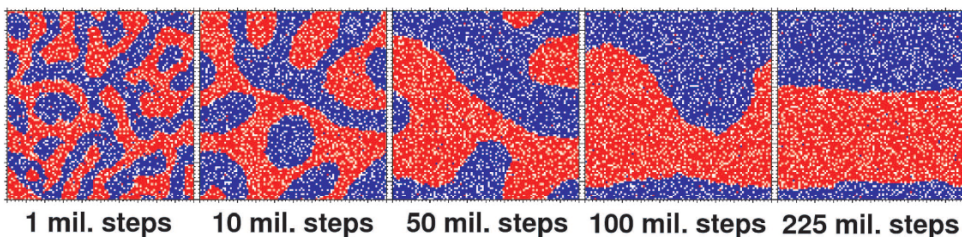


Figure 7.1: The Schelling model of a liquid on a periodic lattice [Vinković/Kirman, 2006, Fig. 5 a]. After 225 mil. steps the agents are fully segregated. The segregation requires many iterations if agents are allowed only to jump to the positions nearest to them.

If a swarm contains a sufficient number of agents, self-organization may emerge. *Self-organization* is defined as the spontaneous formation of patterns by a system itself [Kelso, 1997, p. 8 ff.], without any central control. The snowflake in Figure 7.2 serves as an example of self-organization. During self-organization, novel and coherent structures, patterns, and properties may arise [Goldstein, 1999]. This ability of a system to produce phenomena on a new, higher level is called *emergence* [Ultsch, 1999], and it is separately discussed in the next section.

“Self-organizing swarm behavior relies on four basic ingredients” [Bonabeau et al., 1999, pp. 22-25]: *positive feedback*, *negative feedback*, *amplification of fluctuations* and *multiple interactions*. The first two factors promote the creation of convenient structures and help to stabilize them. Fluctuations are defined to include errors, random movements and task switching. For swarm behavior to emerge, multiple interactions are required. Agents can communicate with each other either directly or indirectly. An example of direct communication is the dancing behavior of bees, in which a bee shares information about a food source, such as how plentiful it is and its direction and distance away [Karaboga/Akay, 2009]. Indirect communication is observed, for example, in the behavior of ants [Schneirla, 1971]. If the agents communicate only through modifications to their environment (through pheromones, for example), then this type of communication is defined as *stigmergy* [Beckers et al., 1994; Grassé, 1959].

The exact number of agents required for self-organization is unknown, but it should be not so large that it must be handled in terms of statistical averages and not so small that it can be treated as a few-body problem [Beni, 2004]. For example, 4096 neurons are required for self-organization in SOMs [Ultsch, 1999], and for the coordinated marching behavior of locusts, a minimum density of least $73.8 \text{ locusts}/m^2$ was reported in [Buhl et al., 2006, p. 1404].



Figure 7.2: Example of self-organization: a large, 10.1x10.1 mm snow crystal [Libbrecht, 2016]. This snowflake is a spontaneous formation of a pattern by molecules of H_2O .

Considering the two requirements stated above, Beni defined a swarm as a formation of cellular robots with a number exceeding 100 [Beni, 2004]. Here, consistent with [Beni, 2004], the argument is made that for self-organization³⁸, the number of agents should be higher than 100. The two main types of swarm-based analysis discussed in data science, namely, particle swarm optimization (PSO) and ant colony optimization (ACO) [Martens et al.], are distinguished by the type of communication used: PSO agents communicate directly, whereas ACO agents communicate through stigmergy. PSO methods are based on the movement strategies of particles [Kennedy/Eberhart, 1995] and typically used as population-based search algorithms [Rana et al., 2011], whereas ACO methods are applied for sorting tasks [Martens et al., 2011]. In addition to being used to solve discrete optimization problems, PSO has been used as a basis for rule-based classification models, e.g., AntMiner, or as an optimizer within other learning algorithms [Martens et al., 2011], whereas ACO has been used primarily for supervised classification within the data mining community [Martens et al., 2011]. Pseudocode for both types of algorithms and illustrative descriptions can be found in [Abraham et al., 2006].

7.1 Artificial Behavior Based on DataBots

The term DataBots refers to agents in the sense discussed here. DataBots were introduced in [Ultsch, 2000a] as the first artificial-behavior-based approach to data science. Each DataBot $b_j \in B$ has a position $i_j \in O$ and takes in food, consumes food and stores quantities of food. Quantities of food are represented by numbers in the range from 0% to 100%. All positions lie on a toroidal lattice, and each DataBot is capable of detecting a scent λ at its current position. This approach is used to perform clustering tasks.

In [Ultsch, 2000c], each DataBot possesses an opinion, defined by one high-dimensional data point, and the DataBots are used as a projection method for a classification task. The movement of the DataBots is defined in terms of probabilities, which are computed using various movement programs called strategies, for each of the four directions (south, east, west and north) and for no movement (origin). With the use of these strategies, self-organization of the system is possible. Unlike in ACO methods, each DataBot possesses an opinion defined by a high-dimensional data point [Ultsch, 2000c]. Hence, reduction of the agents is impossible.

[Kämpf/Ultsch] suggested the use of movement strategies with a decreasing neighborhood radius. The underlying idea of the decreasing radius approach is to promote self-organization, first of a global structure and then of local structures [Kämpf/Ultsch, 2006]. In [Herrmann/Ultsch, 2008b], a set of additional strategies was defined for a subset of DataBots based on labeled data, requiring a prior classification. The authors used this approach to address a classification task by combining it with emergent self-organizing map (ESOM) and the gray-scale two-dimensional U-matrix method. The U-matrix was partitioned into clusters using an entropy-based heuristic algorithm called U*C [Ultsch, 2006]. Here, it is assumed that the DataBots are defined similarly to their definition in [Herrmann/Ultsch, 2008b]: Let each DataBot $b_j \in B$ be an agent identified by a numerical vector $z_j \in \mathbb{R}^d$; it resides on a large, finite, two-dimensional discrete lattice that is embedded on the surface of a torus [Ultsch, 2003a]. The

³⁸ Beni himself only indirectly restricted systems that exhibit self-organization to those consisting of more than 100 agents [Beni, 2004].

current position of DataBot b_j is denoted by $i_j \in O$. Every DataBot $b_j = \{i_j, z_j\}$ emits a scent λ , which is detected by all other DataBots in its neighborhood.

By analyzing ant-based clustering³⁹ (ABC) [Lumer/Faieta, 1994] and the batch self-organizing map (batch-SOM) method [Kohonen/Somervuo, 2002] the local stress of an ABC projection⁴⁰ can be extracted [Herrmann, 2011, pp. 137-138; Herrmann/Ultsch, 2008a, p. 3; 2008c, p. 217; 2009, p. 4]: It is an upper limit on the best matching unit criterion⁴¹ of batch-SOM and forms the topographic term of the Attractiveness function used in ant-based clustering. [Ultsch/Herrmann, 2010] used this mathematical stress term to define a scent as follows:

Let $D(l, j)$ be the distance between two points $x_l, x_j \in I$, let $d(l, j)$ be the corresponding distance in the output space O , and let $h_R: R \rightarrow [0,1]$ be an arbitrary but continuous and monotonically decreasing function; then, the scent $\lambda(b_j, R): \mathbb{R}_0^+ \times O \rightarrow \mathbb{R}_0^+$ is defined as

$$\lambda(b_j, R) = \frac{\sum_{l \in I} h_R(d(j, l)) * D(j, l)}{\sum_{l \in I} h_R(d(j, l))} \quad (7.1)$$

The scent λ is the weighted sum of the distances to neighboring objects; consequently, h_R “realizes a neighborhood function by means of focus” [Herrmann, 2011, p. 65]. To better distinct this neighborhood function from the Databionic swarm, in the following chapters it will be referred to with the same capital letter $F_R = h_R$ as in [Herrmann, 2011].

7.1.1 Swarm-Organized Projection (SOP)

The discussion in this section is based on the thesis of [Herrmann, 2011], which is a continuation of the work of [Herrmann, 2009; Ultsch/Herrmann, 2010]. The SOP algorithm was proposed as a self-adaptive projection method with the aim of creating a cohesive visualization of clusters [Herrmann, 2011]. The algorithm combines a DataBot approach, a scent definition derived from the above analysis of ABC, and Schelling’s segregation model [Schelling, 1969]: the better (weaker) the scent λ becomes, the happier the DataBot is. The SOP algorithm, as presented in Listing 7.1, operates on a finite data set with pairwise dissimilarities, which are usually defined as Euclidean distances [Herrmann, 2011]. The numeric vector z_j associated with each DataBot b_j represents a high-dimensional data point, and the cardinality of the data set I is equal to the number of DataBots. The positions of the DataBots are defined on a rectangular lattice tiling (quad grid) O , which is typically toroidal but could also be planar, in Cartesian coordinates $i(x, y) \in O$, where the numbers of lines L and columns C must be set by the user. Every DataBot chooses between its current position and one new position. If the scent λ , which is defined by the function F_R , would be weaker in its new neighborhood, then the DataBot jumps to the new position. Another DataBot may already be located in the new position, but this does not affect the decision to jump.

In each iteration, all DataBots are allowed to move simultaneously [Herrmann, 2011]. An epoch ends when the following condition is met [Herrmann, 2011]: As long as the number of DataBots that want to jump exceeds an arbitrary threshold value, called a fixed point in [Herrmann, 2011],

³⁹ See next section for a more detailed description.

⁴⁰ In [Herrmann/Ultsch, 2008a] called topographic mapping.

⁴¹ It “is a weighted sum of local input space distances” [Herrmann/Ultsch, 2009, p. 4].

the current epoch proceeds to the next iteration. Otherwise, the next epoch starts, with a decrease in the neighborhood radius R . To ensure the convergence of the algorithm, a maximum number of iterations must be set. [Kohlhof, 2010] proposes a 5% threshold and a maximum number of 500 iterations, but in [Herrmann, 2011], no exact numbers are indicated.

The maximum possible distance in the map space is defined by $R_{max} = \sqrt{L^2 + C^2}$, and the algorithm ends if the smallest possible radius $R = 1$ is reached [Herrmann, 2011, p. 65]. The following contradiction should be taken into account: sometimes, a different minimal radius (e.g., $R=8$ in [Herrmann, 2011, p. 118] for the gene data set, $R>1$ in [Herrmann, 2011, p. 167] for the GPD194 data set) is chosen without any scientific basis other than the author's experience. In practice, the neighborhood function F_R is chosen to be a Gaussian function where the mean is equal to zero and the standard deviation is equal to the radius R . Each possible new position is drawn from a Gaussian-shaped probability distribution (Fig 4.1) [Herrmann, 2011, p. 64]. Pseudocode for the SOP algorithm is provided in [Herrmann, 2011, p. 65], with the scent $\lambda(b_j)$ defined as in equation (1).

Previous work has revealed, based on the practical experience of the inventor [Herrmann, 2009], that SOP is almost as good as or even better than the best of its carefully parameterized competitor methods, such as curvilinear component analysis (CCA), t-distributed stochastic neighbor embedding (t-SNE) and ESOM, in terms of the 1-nearest-neighbor classification accuracy and the specially formulated dispersion measure of [Herrmann, 2011, p. 101] on several natural and artificial data sets.

function $O=sop(l)$

for all $z_i \in I$: *assign an initial random Cartesian position* $i(x,y) \in O$ *on the lattice to generate DataBots* $b_i \in B$

for $R=\{Rmax, \dots, 1\}$ *do*

$m=Gaussian(R)$ *of a Gaussian-shaped distribution:* $N(m(x), s) + N(m(y), s)$

iteration=0

repeat

for $j=\{1, \dots, n\}$ *do*

$l = \operatorname{argmin}_j(\lambda(b_j))$ *with* $j = \{i, m\} \in O$

end for

iteration = *iteration* + 1

until $\{l \in O \text{ fix with } |\{l \in O \mid l = m\}|\} < \text{threshold}\}$ *OR* (*iteration* > i_max)

return O

end function SOP

Listing 7.1: The swarm-organized projection (SOP) algorithm as described in [Herrmann, 2011, p. 65]. The are some parameters to be set by a user (e.g. $Rmax$, $threshold_max$, i_max , ...).

7.2 Swarm Intelligence for Unsupervised Machine Learning

As mentioned earlier in this chapter, there are two main types of artificial swarm optimization methods: PSO and ACO. In unsupervised learning, two additional approaches are known. The first one is based on bees [Karaboga/Akay, 2009], and the second is based on foraging theory [Stephens/Krebs, 1986].

For clustering tasks, PSO has mainly been applied in hybrid algorithms [Esmine et al., 2015]; e.g., [Van der Merwe/Engelbrecht, 2003] applied PSO combined with k-means clustering. Here, it is argued that the hybridization of PSO and k-means may improve the choice of centroids or may, in some special cases, even allow the problem of the number of clusters to be solved. However, this approach is subject to several of the shortcomings of k-means, which is known to search for spherical clusters [Hennig et al., 2015, p. 721]/[Hennig, 2015a, p. 18]; i.e., it is unable to find clusters in elementary data sets, such as those in the Fundamental Clustering Problems Suite⁴² (FCPS) [Ultsch, 2005a].

According to [Rana et al., 2011], the advantages of the clustering process when the PSO approach is used are that it is very fast, simple and easy to understand and implement. “PSO also has very few parameters to adjust [Eberhart et al., 2001] and requires little memory for computation. Unlike other evolutionary and mathematical algorithms it is more computationally effective” [Rana et al., 2011] (citing [Arumugam et al., 2005]). Again according to [Rana et al., 2011], the disadvantages are the “poor quality results when it deals with large and complex data sets”. “PSO gives good results and accuracy for single objective optimization, but for a multi objective problem it becomes stuck in local optima” [Rana et al., 2011] (citing [Li/Xiao, 2008]). Another problem with PSO is its tendency to reach fast and premature convergence at mid-optimum points [Rana et al., 2011]. It is difficult to find the correct stopping criterion for PSO [Bogon, 2013, p. 155], which is usually one of the following: a fixed maximum number of iterations, a maximum number of iterations without improvement or a minimum objective function error [Abraham et al., 2006; Esmine et al., 2015]. Hybrid PSO algorithms usually optimize an objective function [Bogon, 2013, pp. 39 ff, 46] and therefore always make implicit assumptions regarding the underlying structures of the data (see chapters 2, 4 and 5 for details). Notably, there is no single “best” criterion for obtaining a clustering because no precise and workable definition of “a cluster” exists [Jain/Dubes, 1988, p. 91]. For the task of dimensionality reduction, the swarm-inspired projection (SIP) method [Su et al., 2009] are discussed later in this section.

ACO methods for clustering tasks are referred to as ABC methods (for an overview, see [Kaur/Rohil, 2015]). ABC methods model the behavior of ant colonies, and data points are picked up and dropped off accordingly [Bonabeau et al., 1999]. ABC was introduced by [Deneubourg et al., 1991] as a way to explain the phenomenon of the gathering and sorting of corpses observed among ants. In an experiment (Figure 7.3), the ants formed cemeteries of dead ants that had been randomly scattered beforehand. [Deneubourg et al., 1991] proposed probability functions for the picking up and dropping off of the corpses. Because ants are very specialized in their roles, several different types of ants of the same species exist in a colony, and different individuals in the colony perform different tasks. The probabilities are calculated as functions of the number of corpses of the same type in a nearby area (positive feedback).

⁴² See also the results presented in chapter 9.

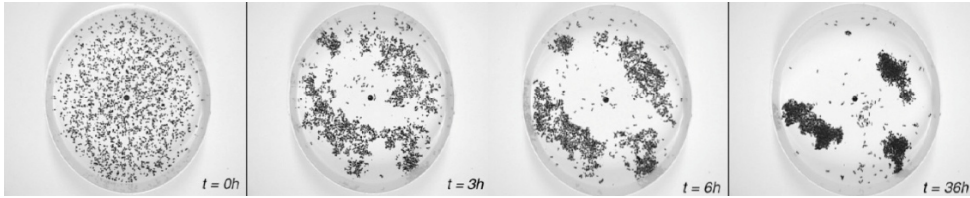


Figure 7.3: Randomly scattered ant corpses are clustered by living ants in a matter of hours [Bonabeau et al., 1999, p. 151; Martens et al., 2011, Fig.5]. The different stage depicted correspond to 0, 3, 6 and 36 hours after the beginning of the experiment.

For a clustering task, the ants and data points (representing ant corpses) are randomly placed on a lattice, and the ants move randomly across the lattice, at times picking up and carrying the data points [Lumer/Faieta, 1994]. The probabilities of picking up and dropping off the data points are modified according to a dissimilarity-based evaluation of the local density (see [Kaur/Rohil, 2015] and [Jafar/Sivakumar, 2010], citing [Lumer/Faieta, 1994]).

[Handl et al., 2006] enhanced the algorithm; they called their version Adaptive Time-dependent Transporter Ants (ATTA) because they incorporated adaptive heterogeneous ants and time-dependent transport activities into the algorithm. Further improvements to the picking up and dropping off activities were presented in [Omar et al., 2013; Ouadfel/Batouche, 2007], and improvements to the initialization and post-processing were proposed in [Aparna/Nair, 2014]. Another version of the approach was developed by introducing an annealing scheme [Tsai et al., 2004]. A feature of ABC algorithms is that the clustering objective is implicitly defined: neither the overall clustering objective nor the type of clusters sought is explicitly defined at any point during the clustering process⁴³ [Handl/Meyer, 2007].

The main problem in ABC lies in the fact that the picking up and dropping off behaviors are independent of the number of agents required to execute the task [Herrmann, 2011, p. 81; Herrmann/Ultsch, 2008a, 2008c, 2009; Tan et al., 2006]. Furthermore, ABC methods can be regarded as derived from the batch-SOM algorithm [Herrmann/Ultsch, 2008a]. From this perspective, an ABC algorithm possesses an objective function, which can be decomposed into an output density term multiplied by one minus a topographic quality term [Herrmann, 2011, pp. 137-138; Herrmann/Ultsch, 2008a, p. 3; 2008c, p. 217; 2009, p. 4]. Both terms are minimized simultaneously [Herrmann/Ultsch, 2008a, 2008c, 2009]. The output density term is easy to optimize but distorts the correct clustering of the data. Here, it is argued that at least 100 agents are required for self-organization in a swarm. However, this many agents are not required in ABC methods, and consequently, the self-organization property of ABC-based swarm algorithms is controversial. Methods of the third type are founded on an analysis of the behavior of bees [Karaboga, 2005]. These are hybrid approaches to clustering that use swarm intelligence in combination with other methods, e.g., k-means⁴⁴ [Karaboga/Ozturk, 2011; Marinakis et al., 2007; Pham et al., 2007; Zou et al., 2010] or SOM [Fathian/Amiri, 2008].

To the best of the author's knowledge, only seven instances of the application of AI in projection methods exist. One method is based on foraging theory, which focuses on two basic prob-

⁴³ This feature will be used in Databionic swarm.

⁴⁴ k-means is known to search for spherical clusters [Hennig et al., 2015, p. 721]/[Hennig, 2015a, p. 18]; see above.

lems: which prey a forager should consume and when a forager should leave a patch [Stephens/Krebs, 1986, p. 6]. A forager is viewed as an agent who compares a potential energy gain with a potential opportunity for finding an item of a superior type [Martens et al., 2011] (citing [Stephens/Krebs, 1986]). This approach is also called the prey model [Martens et al., 2011]: the average energy gain can be mathematically expressed in terms of the expected time, energy intake, encounter rate and attack probability for each type of prey. In the projection method proposed by [Giraldo et al., 2011], in addition to the characteristics of the approach described above, the “foraging landscape was viewed as a discrete space, and objects representing points from the dataset as prey.” There were three agents defined as foragers. Here, the approaches based on the prey model are classified as basic swarm algorithms.

A second method, called the self-organizing swarm (SOSwarm) method, is a clustering method based on a hybrid of PSO and SOM [O’Neill/Brabazon, 2008]. In SOSwarm, 100 particles were used on a 10x10 SOM feature map. However, because only a few units are used, SOSwarm represents a combination of k-means-SOM (see chapter 3) with PSO. Thus, it can be viewed as an application of swarm intelligence, but it is questionable whether this swarm is self-organizing because 4096 neurons are required for self-organization in SOMs [Ultsch, 1999] and the conditions for self-organizing swarm behavior may not apply [Bonabeau et al., 1999, pp. 22-25].

A third method is known as the swarm-inspired projection (SIP [Su et al., 2009], as briefly mentioned above. SIP is a PSO approach that is loosely related to foraging theory because it is inspired by the foraging behavior of doves. The authors report that the number of doves should be significantly smaller than the number of data points and need only be higher than the expected number of clusters. Because of the small number of agents used, it is questionable whether this swarm is self-organizing, but as a PSO approach, it is an example of swarm intelligence.

The fourth approach, SOP [Herrmann, 2011], was already introduced. In terms of swarm behavior, the SOP algorithm does not consider collision avoidance (see the second section of this chapter), as seen from the fact that one or more DataBots may occupy the same position. After an annealing process, the SOP agents are uniformly distributed [Herrmann, 2011, pp. 68-69]; thus, the principle of flock centering is also disregarded. In the next chapter, it will be shown that the SOP algorithm also does not necessarily exhibit the property of *fluctuations* (referred to in the next section as *randomness*) because the position choices of the DataBots are predictable because of their self-interaction and the oblique neighborhood definition. In summary, SOP is a self-organizing swarm of DataBots based on Schelling’s idea to unsupervised machine learning that cannot be regarded as an example of swarm intelligence.

Because ABC methods can be reduced to one ant, these approaches are classified as basic swarms. To exhibit swarm intelligence, a swarm must contain more than one independent agent. Therefore, LF [Lumer/Faieta, 1994] and its derivatives⁴⁵ ATTA-TM [Handl et al., 2006] and ASM [Xu et al., 2007] are not applications of swarm intelligence. Notably, the argument presented here is only valid for ABC methods of unsupervised learning; the categorization may prove invalid for other ACO methods that are supervised.

⁴⁵ The fifth, sixth and seventh applications of unsupervised learning.

The discussion presented in this section is summarized in Figure 7.4, in which only projection methods are explicitly listed. All of the various methods used for clustering cannot be illustrated in one figure. Thus, only general hybrid types are depicted. For all of the publications mentioned above, there is currently no open-source code⁴⁶ available except for applications of rule-based classification [Martens et al., 2011].

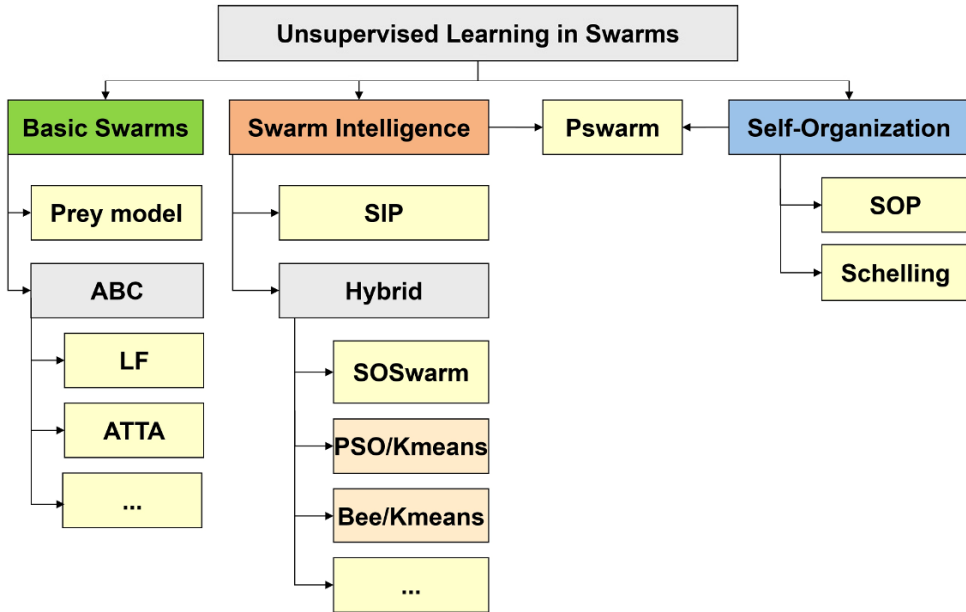


Figure 7.4: Types of swarm algorithms used in unsupervised learning. Pswarm will be introduced in the next chapter; it combines self-organization with swarm intelligence. Various PSO and bee hybrids are used for clustering tasks. Most of these are based on k-means. Aside from Schelling's segregation model, only projection methods are explicitly listed. Abbreviations: ant-based clustering (ABC), particle swarm optimization (PSO).

⁴⁶ The authors of [O'Neill/Brabazon, 2008; Su et al., 2009; Giraldo et al., 2011] were contacted via email, but only Giraldo et al. responded and provided their source code. Due to various limitations, it could not be used for this thesis.

7.3 Missing Links: Emergence and Game Theory

Through self-organization, novel and irreducible⁴⁷ structures, patterns, and properties can emerge in a complex system [Goldstein, 1999]. In analogy to SOMs [Ultsch, 1999], this idiosyncratic behavior of a swarm is defined here as *emergence* (see also [Stephan, 1999]).

Sometimes, a distinction is made between strong and weak emergence [Janich/Duncker, 2011, p. 19]. Here, only strong emergence is relevant. In the literature, the existence of emergence is controversial⁴⁸; it is possible that the concept is only required because the causal explanations for certain phenomena have not yet been found [Janich/Duncker, 2011, p. 23]. Figure 7.5 presents an example of emergence in swarms. The non-deterministic movement of fish is temporarily and structurally unpredictable and consists of many interactions among many agents. Nevertheless, this fish school forms a ball-like formation.

It appears that the concept of emergence has remained unused and rarely discussed in the literature on swarm intelligence, although it is a key concept in AI [Brooks, 1991]. Emergence is mentioned in the literature as a biological aspect of swarms [Garnier et al., 2007], in distributed AI for complex optimization problems [Bogon, 2013, p. 19], in the context of software systems [Bogon, 2013, p. 19] (citing [Timm, 2006]) and as emergent computation [Forest, 1990]. Contrary to Forest, who assumes that only cooperative behavior can lead to emergence [Forest, 1990, p. 8], this works shows that egoistic behavior of a swarm can lead to emergence as well (see chapter 8). With regard to swarms, emergence should be a key concept. The four factors leading to emergence in swarms are

- I. Randomness
- II. Temporal and structural unpredictability
- III. Multiple **non-linear** interactions among **many** agents
- IV. Irreducibility

[Bonabeau et al., 1999, p. 23] agrees with [Ultsch, 1999, 2007] regarding the first factor: “*Randomness* is often crucial, since it enables the discovery of new solutions, and *fluctuations* can act as seeds from which structures nucleate and grow.” Here, an algorithm is considered to have the property of *randomness* if it uses a source of random numbers in its calculations (non-determinism) [Ultsch, 2007]. The power of randomness is evident in Schelling’s segregation model (Fig 3.).

The second factor, *unpredictability* [Ultsch, 2007, O’Connor/Wong, 2015], is incompatible with the PSO approach, in which an objective function is optimized [Martens et al., 2011] and, therefore, predictable assumptions are implicitly made regarding the structures of data sets in the case of unsupervised machine learning (see chapter 4 for further details on projection methods).

The third factor, multiple interactions among many agents, was identified by [Forest, 1990, pp. 1-2] for nonlinear systems. Although [Bonabeau et al., 1999] defines a requirement of multiple interactions for self-organization, the authors argue on page 24 that a single agent may also be sufficient. This is not the case for emergence, for which many elementary processes are mandatory [Beni, 2004; Ultsch, 1999]. Hence, ACO methods cannot exhibit the property of emergence. Nonlinearity means that adding or removing interactions among agents or any agents

⁴⁷ There is no way to derive the property from any part, subset or partial structure of the system [Ultsch, 2007].

⁴⁸ For applications, the existence of emergence is irrelevant. Even if emergent phenomena can be causally explained, they can still be used in the future (see [Stephan, 1999] for discussion).

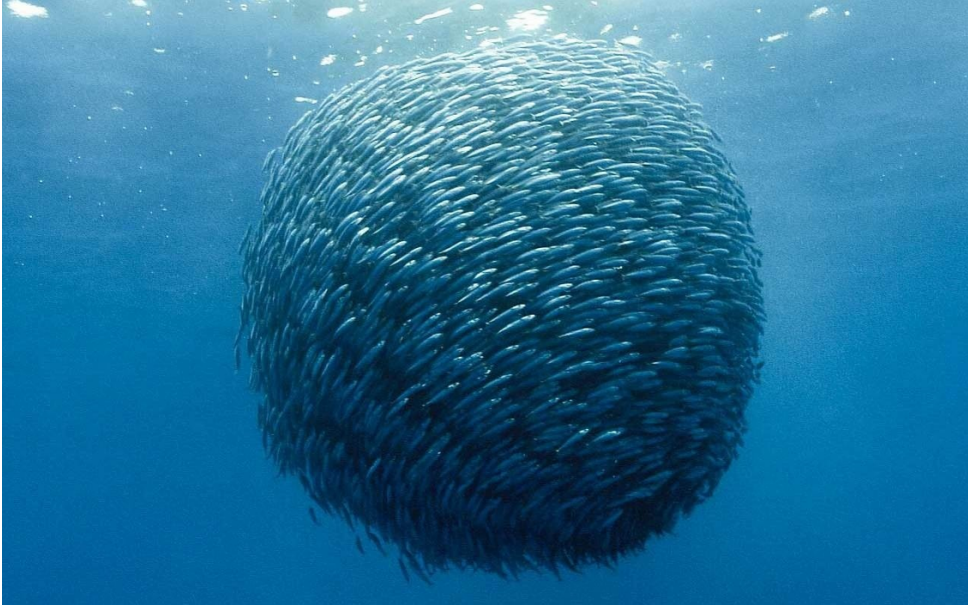


Figure 7.5: A fish swarm in the form of a ball [Uber_Pix, 2015]: an example of emergence in swarms. It illustrates the ability of a system to produce phenomena on a new, higher level.

themselves results in behavior that is linearly unpredictable. For example, the removal of one DataBot results in the elimination of one data point.

The fourth factor, *Irreducibility* [Kim, 2006, p. 555, Ultsch, 2007, O'Connor/Wong, 2015], means that the (novel) property cannot be derived from any agent (or part) of the system, but is only a property of the whole system. It is the ability of a system to produce phenomena on a new, higher level [Ultsch, 1999]. Vividly, it mark a distinction between the self-organization in Figure 7.2, where essentially a pattern of a snow flake could be derived by the physical properties and chemical bonds of H_2O and Figure 7.5, where the formation of a ball cannot be predicted from any fish itself.

The second missing link is a connection to game theory, in which the four axioms of self-organization — *positive* and *negative feedback*, *amplification of fluctuations* and *multiple interactions* — are apparent. Game theory was introduced by [Neumann/Morgenstern] in 1947. The purpose of game theory is to model situations⁴⁹ in which multiple players interact with each other or affect each other's outcomes [Nisan et al., 2007, p. 3] (*multiple interactions*). Here, the focus lies on a general, not zero-sum, n-person game [Neumann/Morgenstern, 1953, p. 85]. A game is defined as a scenario with n players $i=1, \dots, n$ in which each player makes a choice [Neumann/Morgenstern, 1953, p. 84] (*amplification of fluctuations*⁵⁰).

Let a game G be defined by n players associated with n non-empty sets Π_1, \dots, Π_n , where every set Π_i represent all choices made by player i ; then, the pay-off function is defined as

$$p = (p_1, \dots, p_n): \Pi_1 \times \dots \times \Pi_n \rightarrow \mathbb{R}^n \quad (7.2)$$

⁴⁹ To be more specific, rational decision-making behavior in social conflict situations.

⁵⁰ Task switching.

The choices of each player determine the outcome for each player, and the outcome will, in general, be different for different players [Nisan et al., 2007, p. 9]. In a game, the payoff for each player depends on not only his own choices but also the choices of all other players [Nisan et al., 2007, p. 9] (*positive* and *negative feedback*). Often, the choices are defined based on a set of mixed strategies for each player. From the biological point of view, these mixed strategies may include the five main principles of collective behavior: *Homogeneity*, *Locality*, *Velocity Matching*, *Collision Avoidance*, and *Flock Centering* [Grosan et al., 2006].

In a game with n players, let the k choices of player i be defined by a set $\Pi_i = \{\pi_1^i, \dots, \pi_\alpha^i, \dots, \pi_k^i\}$, where π_α^i indicates the i^{th} player's α^{th} choice; then, a mixed strategy $s_j(i) \in S_i$ for player i is defined by

$$s_j(i) = \sum_{\alpha=1}^{k(i)} c_\alpha(i) \pi_\alpha(i) \quad (7.3),$$

where $\sum_{\alpha=1}^{k(i)} c_\alpha(i) = 1$ and all $c_\alpha(i) \geq 0$.

For noncooperative games, [Nash, 1951] proved the existence of at least one equilibrium point. Let $t_j(i) \in S_i$ be the mixed strategy that maximizes the payoff for player i ; then, the Nash equilibrium is defined as

$$p_i(s(1), \dots, s(i-1), t_j(i), s(i+1), \dots, s(n)) = \max_{t_j(i) \in S_i} p_i(s(1), \dots, s(n)) \quad (7.4)$$

if and only if this equation holds for every i [Nash, 1951]. The mixed strategy $t_j(i) \in S_i$ is the equilibrium point if no deviation in strategy by any single person results in a greater profit for that person. A Nash equilibrium is called *weak* if multiple mixed strategies $t_j(i) \in S_i$ for the same person exist in equation (4) that result in the same maximal payoff p_i , whereas in a *strong* Nash equilibrium, even a coalition of players cannot further increase their payoffs by simultaneously changing their strategies $t_j(i) \in S_i, i = 1 \dots m \leq n$, in (4). An illustrative example is the prisoner's dilemma [Poundstone, 1992]. Because of the interactions among the mixed strategies of all players that govern the payoff for a single player, the Nash equilibrium is not necessarily unique, and multiple different equilibria could exist.

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

