Emergence in Problem Solving, Classification and Machine Learning

Joël Quinqueton

Emergence in Problem Solving, Classification and Machine Learning

Joël Quinqueton

LIRMM

Computer Science Department

161 rue Ada, F34392 Montpellier, france

jq@lirmm.fr

Abstract

Emergence is usually the way in which a collective organisation behaves differently than the sum of its elements. We propose here an overview of different ways this paradgm is used in several fields of Artificial Intelligence and we propose some theoretical tracks relying on some works in thye field of Machine Learning.

1 Introduction

Emergence is usually used to denote a collective phenomenon in which the whole organization behaves differently than the sum of its elements. We shall in this paper present an overview of different ways this paradigm is used in Artificial Intelligence. Then, we give some theoretical tracks. The early attempts of philosophers to propose a theory of emergence lead to point out 3 propositions [8]:

1. that there are levels of existence defined in terms of degrees of integration;

2. that there are marks which distinguish these levels from one another over and above the degrees of integration;

3. that it is impossible to deduce the marks of a higher level from those of a lower level, and perhaps also (though this is not clear) impossible to deduce marks of a lower level from those of a higher.

Emergence, as it appears in the previous proposed definitions, underlies several levels. Between these levels, a fixed point appears, called autopoiesis [10], in which the upper level appears from the lower one, but this upper level has a causal link to the lower one. Examples of such a behaviour can be found in simulations of termites or ants, or also in problem solving with ants.

2 Emergence in nature

2.1 A small example

As a first example, let us look at the artificial termites behaviour, as simulated in Starlogo [1]. The interest of this sim-
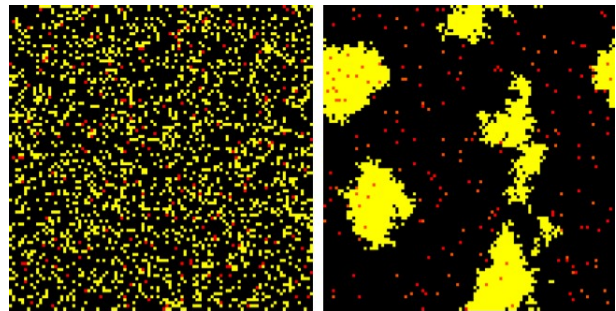

Figure 1. The termites simulation in Starlogo

ulation is first that it is based on very simple agents, and second that it exhibits a significant collective behaviour. The environment is a flat black landscape with yellow pieces of wood on the ground. Each agent, called a termite, moves randomly in the environment, as long as there is nothing on the ground (black points). When there is a piece of wood on the ground (yellow points), two cases are possible:

- the agent holds nothing : it takes the piece of wood and continue walking randomly with the piece of wood

- it holds a piece of wood: it looks for a free (black) place around to put the piece of wood.

The pieces of wood are first randomly distributed, like in the left part of figure 1. If we wait for enough time, all the pieces of wood will be connected, and make what we call a pile: such an almost final state is shown on th right of the figure.

---

[1]Open simulation platform available at `http://education.mit.edu/starlogo/`

Let us now explain why this happens. A new pile necessarily results from the division of an existing one, it can never start from scratch because of the last rule. Such a division is less probable as the size of the pile increase. Then, the number of piles decrease over time. As it is finite and at least one, then it is one after a finite time. There is no "protection" of existing piles, otherwise it would lead to a lot of small piles instead of a single big one.

The concept of pile is necessary for this explanation, and the resulting the causal model involves it as central. Then, this concept appeared from the observation and has a causal relation with the pieces of wood. This top-down causal influence is characteristic of emergence.

2.2 Swarm Intelligence

There are a lot of examples of Swarm Intelligence in nature, the most famous of whch is certainly the way ants behave to find food. The usual features which are considered to be characteristic of this kind of collective intelligence are:

- multiple autonomous agents rather than a single agent

- involve local sensing and actions rather than global ones

- the control is distributed rather than centralized

- simple rules define the individual behaviour instead of complex ones

- the organization is bottom up rather than top down

- the system behaviour is emergent rather than pre-programmed

These characterisctic properties can be illustrated on the example of ants, starting with the famous experiment of the binary bridge. In this experiment, a group of ants generally find the shortest path from a nest to a food source and back. This is accomplished by communicating with each other not directly, but through the environment using pheromones. Such a way in which the work already done drives the work to be done is called stigmergy.

Initially at the choice point between two paths, an ant makes a random decision with a probability of .5 of turning either left or right. Two ants leave nest by different paths, two get food and return to nest by different paths Once ant on shortest path returns, there is almost twice as much pheromone on that path near the nest. As other ants follow the shorter path, the pheromone trail is reinforced causing a positive feedback effect. As less ants follow the longer path, the pheromone dissipates and still fewer ants take that path producing a negative pheromone effect.

Another key feature of ant colonies is their ability to adapt "spontaneously" to changes in the environment. For

Figure 2. The binary bridge experiment

instance, when a stone falls in the middle of a straight path built by an ant colony, the ants find the shortest way to bypass the obstacle, precisely by applying the previous behaviour.

If we try to generalize these experiments, we obtain three key aspects:

- Positive Feedback (auto catalytic)

- The pheromone track is stronger (faster rebuilt) on the shortest path

- Every ant walks approximately at the same speed

3 Application to combinatorial problems

Let us look now at the way such principles can be applied to combinatorial problems.

3.1 Ant algorithms

A large amount of work has been driven on the family of Ant Algorithms, both experimental [2] and theoretical [1]. We can illustrate these aspects on a simple combinatorial problem, illustrated on figure 3.

In their theoretical survey, Dorigo and Blum pointed out that, while there are a lot of results about what can be done with these methods, the proofs do no say anything about the time required to find a solution.

Then, Frank Neuman and Carsten Witt [7] propose to apply to compare ACO algorithms to evolutionary compu-
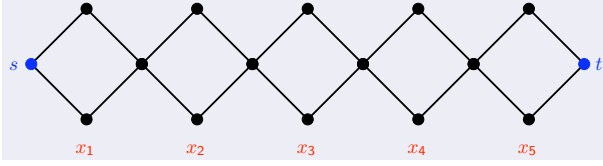
Figure 3. The construction of the graph for 1-ANT on the ONEMAX problem

tation for simple problems such as ONE-MAX, in order to obtain results on convergence speed.

This problem is to find a maximum for the pseudo-Boolean function f : $\{0,1\}^n \rightarrow R$. The Ant algorithm, on this simple problem, consist for each ant to choose among the 2 possible paths at each node. Then, pheromone is updated according to an evaporation factor ρ, which is very important, as if it disappears, then the ants will continue to walk at random.

The way the pheromone track is updated is:

- for used edges $\tau(e) := Min\{\frac{(1-\rho)\tau(e)+\rho}{1-\rho+2n\rho}, \frac{n-1}{2n^2}\}$,

- for unused edges $\tau(e) := Max\{\frac{(1-\rho)\tau(e)}{1-\rho+2n\rho}, \frac{1}{2n^2}\}$

We can see that, if $\rho = 0$, then the values of $\tau(e)$ remains the same, and the ants continue to choose at random. Then ρ appears to have here a similar role to that of cossover operator in evolutionary algorithms. The convergence results are therefor very similar.

3.2   Emergent Algorithms

These algorithms [3] are based on local and decentralized computing. In the example of routing problem, such an algorithm guarantees a strict increasing of the similarity (common bits) between current address and destination. Each agent has a local routing table whose size is $log_2(N)$, where N is the number of bits in the agent's address. The row i of this table contains the path from the agent's address to an address (preferably, but not necessarily, the nearest) which has the same i left bits.

For any current node address c, the following algorithm is applied, given a destination address, d and a path $p = p_1, p_2, ...p_n$:
If ($c = d$) then done
else if ($n > 0$) then
Route d and the path $p_2, p_3, ..p_n$ to c's immediate neighbor p_1.
else if ($n = 0$) then
Let j = number of leftmost bits in common between d and c.
Let p = the path in the $(j+1)^{th}$ entry in c's routing table;

Route d and the path $p_2, p_3, ..p_n$ to c's immediate neighbor p_1.

We can see easily that this algorithm guarantees a strict increasing of the similarity (common bits) between current address and destination

In the domaine Graph Coloring there is also a classical local heuristic where each node take the smallest color not taken by any of its neighbours. It usually leads to a good solution. In Graph Coloring with Ants, pieces of good solutions are assembled to produce the final one.

3.3   Emerging mind

In terms of memory organisation: towards emergent memory A.I. models conscious knowledge and reasoning. The unconscious knowledge is emergent

- at the micro level (psychology)

- at the macro level (economy)

In Multi-agent systems, it corresponds to the collective behavior.

About the emerging mind of the machine, John Cameron Glasgow [4] pointed out that " any implementation of humanlike intelligence in a machine will have to include lower levels, and that the mechanism of that implementation will have to be emergence rather than construction."

In a recent work, we studied a language abnd system, Uniscript [6], designed for storing situated (time and space) facts and events. A concept of *surprise* appeared, when an event leads to a deep reorganization of the memory.

4   Machine Learning and Theory of Emergence

We pointed out in the beginning of this paper the causal relation between the emergent upper level and the lower one. This suggests a glance at the Machine Learning mechanisms which have in common the fact that what is learned has a causal relation to the examples, as it is supposed to be a generalization.

4.1   Possible Worlds Semantics

This Semantics starts from a language L, a set of worlds W and a binary relation $R \subseteq W \times W$.

The language L is supposed to be made of atomic formula, and the 2 operators $\neg$ and $\rightarrow$. The set W of possible worlds with the R relation is usually called the frame $F = (W, R)$. The R relation is called the accessibility one or the influence one. It can be interpreted as the fact that the worlds are similar, for instance, or that an action exists

to go from one world to the other, or that one world follows the other (in the temporal sense, for instance).

In this framework, a model M is a triple (L, F, V) where:

- V is a mapping of the formula in L onto the subsets of W, $V : L \mapsto 2^W$,

- $V(f \rightarrow g) = V(g) \cup \overline{V(f)}$

In the multiagent case, we have one relation for each agent, and we define a new operator in L: $c(J, \alpha)$, which means "Agent J believes that α". The usual definition of such a belief is: $s \in V(c(J, \alpha))$ iff $\forall t; sRt, t \in V(\alpha)$.

Such a belief can have various interpretations, such as the knowledge that any action of J makes α true, or that, in the future, α will always be true, or, in operational context, that when the program J end, α is true, or that α can be prooved by J, for instance.

The necessitation rule, which says that when something is true, then it is necessary (or believed) can be stated as follows: $V(\alpha) \subseteq V(c(J, \alpha))$. As we shall see later, this rule is not desirable in multiagent systems.

Goldblatt's theorem prooves the equivalence of 14 possible properties of the R relation and 14 possible tautologies (true in any interpreation) of the corresponding logics [5].

Among them, we are interested in $(K), (T)$ and (U), then in $(4), (D), (5)$, and (D_c). In fact, (K) is derived from the definitions and does not require any property of the R relation.

For instance, an agent following (T) only believes what is true, while (U) leads to a "confident" agent: it believes that what it believes is true, $c(J, c(J, \alpha) \rightarrow \alpha)$.

4.2 Machine Learning

Some particular cases of R relation are interesting as a framework for Machine Learning:

- If R is empty, the corresponding agent is called a *Probe*. Such an agent believes everything and the opposite, because all Goldblatt's tautologies are true but (T) and (D).

- If R is full, all tautologies but $(D_c$ hold, and the corresponding agent is called a *Master*.

- If R is reduced to the loops, then each world accesses itself, and all tautologies are true. Moreover, such an agent follows the necessity rule. Then all what such an agent believes is true, and conversely. It is called an *Oracle*, and its role in Learning is to provide the examples.

An *Apprentice* is an agent able to learn a formulae β from a given set E of examples of a concept to find, say γ. We suppose that $E \subseteq V(\gamma) \subseteq W$.

For a formulae β learned by our apprentice, we define $R_\beta ::= V(\beta) \times (E \cap V(\beta))$.

An agent A with this accessibility relation checks $(U), (4), (5)$, and we have $V(\beta) \subseteq V(c(A, \gamma))$, then $c(A, \beta \rightarrow \gamma)$ is a tautology. This is close to a what is expected from an apprentice.

In the general case, we have both positive and negative examples, given as 2 sets E^+ and E^-. We suppose now that $E^+ \subseteq V(\gamma) \subseteq W$ and $E^- \subseteq \overline{V(\gamma)} \subseteq W$.

Our apprentice has learned a formulae β and we build

$$R_\beta ::= V(\beta) \times (E^+ \cap V(\beta)) \cup \overline{V(\beta)} \times (E^- \cap \overline{V(\beta)})$$

The minimal required properties for R_β not to be empty is that $E^+ \cap V(\beta) \neq \emptyset$ and $E^- \cap \overline{V(\beta)} \neq \emptyset$, that is, β must be true for at least one positive example and false for at least one negative example. This is a reasonable assumption.

As $V(\beta) \cup \overline{V(\beta)} = W$, the R_β relation is serial, then the belief of our apprentice has no contradiction (the (D) tautology).

4.3 A learning scheme without counter examples

Some learning problems have intrinsically no negative examples. For instance, the prediction of earthquake prone area can only be done by taking as examples the places where an earthquake already occured. Similarly, undesirable web sites (for children) can be learned using examples of bad sites.

More recently, we investigate a way to learn models for constraint programming [9], starting from examples of past solutions. For a timetable problem, for instance, we take the solutions of past years, and look for regular patterns to learn a good viewpoint (variables and domains) to state the constraints of the problem.

In such a case, we can state our learning problem as a family of classical learning problems. For each attribute describing the examples, we split the set into positive examples (the attribute is true) and negative examples (it is false). Then, we learn to predict this attribute from the others.

A suitable property for such a generalisation operator G is $G(G(X)) = G(X)$ for a set of examples X.

5 Conclusion

We have proposed a description of learning mechanisms in such a way that they can lead to a theory of emergence, by building a visibility relationship between possible worlds.

Then, learning without counterexamples appear as a good candidate for a formalization of Emergence. We continue to inverstigate this direction and try to make links with boolean networks.

Appendix   We remind here the correspondance defined by Goldblatt's theorem, for the properties we are interested in. For the others, the reader can refer to Goldblatt's report [5].

The (K) axiom (distribution rule) holds for any relation R:

$$c(a_i, f \rightarrow g) \rightarrow (c(a_i, f) \rightarrow c(a_i, g))$$

Property of the relation	Tautology
Reflexive	knowledge (T)
$\forall s(sR_i s)$	$c(a_i, f) \rightarrow f$
Weakly reflexive	confidence (U)
$\forall s(tR_i s \Rightarrow sR_i s)$	$c(a_i, c(a_i, f) \rightarrow f)$
Serial	non contradiction (D)
$\forall s \exists t(sR_i t)$	$c(a_i, f) \rightarrow \neg c(a_i, \neg f)$
Transitive	positiveintrospection (4)
$\forall s \forall t \forall u(sR_i t \wedge tR_i u \Rightarrow sR_i u)$	$c(a_i, f) \rightarrow c(a_i, c(a_i, f))$
Euclidian	negative introspection (5)
$\forall s \forall t \forall u(sR_i t \wedge sR_i u \Rightarrow tR_i u)$	$c(a_i, f) \rightarrow c(a_i, \neg c(a_i, \neg f)$
Pseudo-functional	(Dc)
$\forall s \forall t \forall u(sR_i t \wedge sR_i u \Rightarrow t = u)$	$\neg c(a_i, \neg f) \rightarrow c(a_i, f)$

References

[1] M. Dorigo and C. Blum. Ant colony optimization theory: a survey. *Theoretical Computer Science*, 344(2-3):243–278, 2005.

[2] M. Dorigo and T. Stützle. *Ant Colony Optimization*. MIT-Press, 2004.

[3] D. A. Fisher and H. F. Lipson. A new method for enhancing survivability in unbounded systems. In *IEEE proceedings*, 1999.

[4] J. C. I. Glasgow. The emerging mind of the machine, 1989.

[5] R. Goldblatt. *Logics of Time and Computation*. Number 7 in CSLI Lecture Notes. CSLI - Center for the Study of Language and Information, Stanford University, 1986.

[6] A. Kiss and J. Quinqueton. Uniscript: a model for persistent and incremental knowledge storage. In *Carpe'04 proceedings*, pages 66–73, New York, USA, october 2004. ACM Multimedia.

[7] F. Neumann and C. Witt. Runtime analysis of a simple ant colony optimization algorithm. In D. V. Arnold, T. Jansen, M. D. Vose, and J. E. Rowe, editors, *Theory of Evolutionary Algorithms*, number 06061 in Dagstuhl Seminar Proceedings. Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany, 2006.

[8] S. C. Pepper. Emergence. *Journal of Philosophy*, 23:241–245, 1926.

[9] G. Raymond, J. Quinqueton, and C. Bessière. Mining historical data to build constraints viewpoints. In *CP'06 modelling workshop*, 2006.

[10] F. J. Varela, H. R. Maturana, and R. Uribe. Autopoiesis: The organization of living systems, its characterization and a model. *Biosystems*, 5(4):187–196, may 1974.