EVOLUTIONARY ALGORITHMS TO SIMULATE REAL CONDITIONS IN ARTIFICIAL INTELLIGENCE AS BASIS FOR MATHEMATICAL FUZZY CLUSTERING

Ness, S. C. C

Evocell Institute, Austria

ABSTRACT

In present-day physics we may assume space as a perfect continuum describable by discrete mathematics or a set of discrete elements described by a programmed probabilistic process or find alternative models that grasp real conditions better as they more closely simulate real behaviour. Clustering logic based on evolutionary algorithms is able to give meaning to unlimited amounts of data that enterprises generate and that contain valuable hidden knowledge. Evolutionary algorithms are useful to make sense of this hidden knowledge, as they are very close to nature and the mind. However, most known applications of evolutionary algorithms cluster data points to one group, thereby leaving key aspects to understand the data out and thus hardening simulations of biological processes. Fuzzy clustering methods divide data points into groups based on item similarity and detects patterns between items in a set, whereby data points can belong to more than one group. Evolutionary algorithm fuzzy clustering inspired multivariate mechanism allows for changes at each iteration of the algorithm and improves performance from one feature to another and from one cluster to another. It is applicable to real life objects that are neither circular nor elliptical and thereby allows for clusters of any predefined shape.

In this paper we explain the philosophical concept of evolutionary algorithms for production of fuzzy clustering methods that produce good quality of clustering in the fields of virtual reality, augmented reality and gaming applications and in industrial manufacturing, robotic assistants, product development, law and forensics as well as parameterless body model extraction from CCTV camera images.

KEYWORDS

Artificial Evolution, Artificial Intelligence, Biology, Big Data, Cellular Automata, Data Interpretation and Analytics, Deep Learning, Features Selection, Genetic Algorithms, Generative Models, Machine Learning, Pattern Recognition, Robotic Process Automation,Simulation, Smart Systems, Virtual Machines, Visualization.

1. FULL PAPER

In order to develop an ultimate model for the universe, the first step is to think about the nature of space. In present-day physics we may assume space as a perfect continuum describable by discrete mathematics or a set of discrete elements described by a programmed probabilistic process or find alternative models that grasp real conditions better as they more closely simulate real behaviour.

Philosophers have been known since ancient times to think deeply about the world around them. Everything they see, be it the stars in the night, the kebab in a Berlin snack bar or the naked

Dhinaharan Nagamalai et al. (Eds) : CSE, AI&FL, CCNET, DBDM, CSIP, CICS - 2019
pp. 01-05, 2019. © CS & IT-CSCP 2019 DOI: 10.5121/csit.2019.90501

breasts of a young sun worshipper, fills your mind with deep philosophical thoughts. One day a philosopher looks at the lawn in front of his kitchen. As he takes a closer look at a small part of the ground with its countless small blades of grass, he begins to doubt the concept of the lawn itself. Is what he sees really the lawn or does he see a single blade of grass and a single blade of grass and a single blade of grass? It's not about counting the individual blades of grass, the number has no meaning. It is important to see the plants with a single glance, each in its own particularity, its own character and its own difference. And not just to see them, to think them. The difficult thing, the philosopher states, is to grasp the whole as the sum of its parts. The philosopher is convinced that the lawn can only really be understood as a collection of blades of grass.

Our philosopher here is subject to a philosophical direction called reductionism, and this program is certainly seductive:

This reductionism is mostly motivated by the fact that people are impressed by the explanatory success of modern science.

Many theories since the Greek philosophers have shown that reduction is possible in many previously unexplained areas. It should therefore be assumed that reductions are also possible in previously unexplained areas. This is called the inductive argument for reductionism: if one does not understand something, one should break it down into its pieces, reduce it to its individual components. Since they are simpler than the whole, one has a greater chance of understanding these parts. And once you have understood them, you put everything back together again.
The question is: can we understand the concept of information through this particle-based reductionism? Does the investigation of a basic unit help us to uncover the true nature of information? Can we learn or simulate something by following the path postulated by Claude E. Shannon in the last century? Can we even divide the information itself into a measure that reflects its information content?

Although very widespread in computer science, arguments for this theory are not based on the history of science, but on considerations of causality. In classical argumentation there are causes for an event on different levels. For example, when a person takes a headache tablet. Then one can indicate different causes for this event.
Possible causes are, for example:

1. a mental explanation, such as the sensation of headache
2. biological processes that triggered certain muscle contractions
3.microphysical processes that cause other microphysical processes that realize tablet swallowing.

The fact that there can be several causes individually or together is a problem insofar as it is unclear whether these causes are independent of each other. After all, there is such a variety of causes in every action and it would be surprising if all these actions constantly have several independent causes. Proponents of reductionism like to argue that headaches are nothing more than a biological process and that every biological process is nothing more than a microphysical process or vice versa. If, however, one accepts multiple causes in solving the problem, one would also have to accept reductionism, since headaches are ultimately identified with a microphysical process.

This centuries-old conceptual reduction is very far removed from nature, i.e. just as far as it abstracts nature.

In the last century, Alan Turing (1912 to 1954) laid firm foundations, including the theoretical basis for universal calculating machines and the idea of artificial intelligence.

In 1936, he presented an abstract description of the solution to mathematical problems, which made him famous: "On Computable Numbers with an Application to the Decision Problem". The decision problem was a fundamental question of mathematics at the time. Put simply, it was about whether an algorithm could automatically find out that a mathematical statement is wrong or correct within a certain framework. Turing conceived - as a thought experiment - a machine with paper stiffeners as a storage medium, a kind of mechanized arithmetic artist: the so-called Turing machine. This computability model is one of the foundations of theoretical computer science today.

Most modern computers are the offspring of a computer built in 1948 based on Turing concepts in terms of their logic architecture.

Nevertheless, most artificial intelligences today are not based on advanced considerations, but on reductions that are based on statistical parameters, so that they can also be calculated with an ordinary calculator.

After all, Turing wanted to crack codes based on human logic in the Second World War. We want to simulate nature and thereby understand it.

How can and must growth and shape be so that today we can use them to simulate nature with a mathematical model? How can we describe morphogenesis, i.e. the development of organisms and organs, using mathematical models in individual areas of application?How does a small structure such as a blade of grass develop into a complex organism such as a lawn?

We see evolutionary algorithms in the field of artificial intelligence today as a class of stochastic, metaheuristic optimization techniques whose operational mode is inspired by the evolution of natural organisms. In those evolutionary algorithms we do not only use genetic programming, ie. the automatic creation or modification of computer programs using heuristic search algorithms to solve optimization and simulation by random selection, combination and variation of the desired parameters. Evolutionary programming of algorithms is different both in the sense that the structure of the program is constant so that only numerical values will change and in the sense that in the next generation only positive mutations are passed on.

As solutions for a certain problem are artificially evolved, those evolutionary algorithms are nature-analogous optimization methods. They show how learning and evolution interact, model ecosystems, immune systems, cognitive systems and social systems and thus useful to model artificial life using mathematics. Evolutionary algorithms do not usually find the best solution for a given problem if successful due to the assigned stochastic and metaheuristic algorithms. But if unsuccessful, they do find a sufficiently good solution, which is desirable in practice, especially for NP complete problems due to the number of mutations and because other NP complete problems are solved by mapping them onto the canonical problem. The methods of different evolutionary algorithms differ from each other primarily in selection, recombinations of the operators that were used, the problem representation and the kind of mapping applied. In evolutionary algorithms, complex functions evolve by building on simpler functions evolved previously, based on previous selection. In this scheme, the first genotypes able to perform a set number of tasks differ from their non-performing parents only by one or two or three mutations. But they differ from starting point of consideration ancestors by many mutations crucial to the new functionality. Complex useful functions can originate by random mutation and selection of a population and even deleterious mutations can serve as stepping-stones in the evolution of ground-breaking complex features.

The thinking model behind an evolutionary algorithm can be represented schematically as follows:

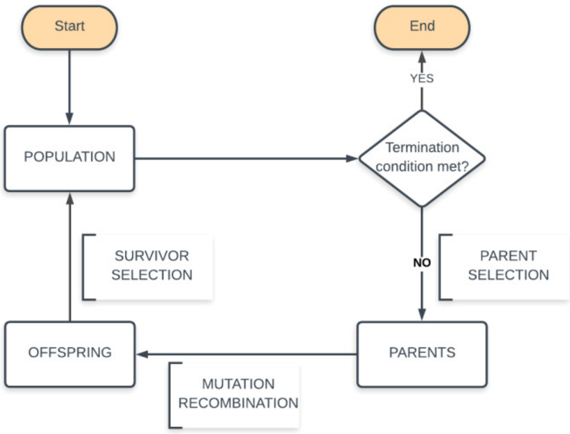

Figure 1: Schematic representation of evolutionary algorithms

The procedure of reproduction in evolutionary algorithm matches real world mutation amid generations of evolutions. The reproducibility of a solution to a given problem in evolutionary algorithms depends completely on what sort of arrangements you have and what sort of issue you wish to comprehend. In the mating capacity there is a variation operator which applies a mutation to the subsequent child solution and allows the next batch of solutions to “mutate” new features that may be superior to the features of the previous generation of solutions. In survivor selection upon incorporating the new functions there is a recognition among individual solutions dependent on their quality. The evolutionary algorithm selects the top performers of the solutions in this generation in a pre-set manner. Survivor selection iterates over many generations of evolution offspring and gives insights when the optimal fitness level is met by a candidate solution in the offspring pool.

Evolutionary algorithms are useful in order to map real life continuous natural selection processes: One aspect is that they are used in case of large or complex data, where regular algorithms need too much time and therefore stability is improved by an evolutionary approach. Another aspect is that biology as evolutionary algorithms help observe how a population looks like in generations. Artificial system inspired algorithms serve as models of living systems for the investigation of open questions in biology. Artificial life studies may help to understand open questions in understanding biological processes, including the origin of life, self-organization, cultural evolution, origin and maintenance of sex, balance in evolution, relations between fitness and adaptedness, structures of ecosystems and the nature of mind.

To sum it up, evolutionary algorithms help solve optimization and design problems by building solutions more fit relative to desired properties, thus leading to a computational evolution useful in the science of artificial intelligence by computers.

REFERENCES:

[1] Banzhaf, Wolfgang &Beslon, Guillaume & Christensen, Steffen & Foster, James & Képès, François &Lefort, Virginie & Miller, Julian &Radman, Miroslav & Ramsden, Jeremy. (2006). From artificial evolution to computational evolution: a research agenda. Nat. Rev. Genet. 7, 729-735. Nature reviews. Genetics. 7. 729-35. 10.1038/nrg1921.

[2] Kenneth A. De Jong and William M. Spears. (1989).Using genetic algorithms to solve NP-complete problems. In Proceedings of the third international conference on Genetic algorithms, J. David Schaffer (Ed.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 124-132.

[3] Lenski, Richard & Ofria, Charles & Pennock, Robert & Adami, Christoph. (2003). The Evolutionary Origin of Complex Features. Nature. 423. 139-44. 10.1038/nature01568.

[4] Miconi, Thomas & and Channon, Alastair. (2005). Analysing co-evolution among artificial 3d creatures. In Proceedings of the 7th international conference on Artificial Evolution (EA'05), El-GhazaliTalbi, Pierre Liardet, Pierre Collet, Evelyne Lutton, and Marc Schoenauer (Eds.). Springer-Verlag, Berlin, Heidelberg, 167-178. DOI=http://dx.doi.org/10.1007/11740698_15

[5] Mitchel, Melanie & Forrest, Stephanie. (1991). Genetic Algorithms and Artificial Life, Artificial Life Vol 1. Num. 3, p 267-289

[6] Papademetriou, Christos. (2012). TO WHAT EXTEND IS THE TURING TEST STILL IMPORTANT?. Pliroforiki Cyprus Computer Society. 28-32.

[7] Singhal, Geet&Panwar, Shipra& Jain, Kanika& Banga, Devender. (2013). A Comparative Study of Data Clustering Algorithms. International Journal of Computer Applications. 83. 41-46. 10.5120/14528-2927.

[8] Taylor, Charles E.&Jefferson, David. (1994). Artificial Life as a Tool for Biological Inquiry. Artificial Life, Volume 1, Number 1-2, Fall 1993/Winter 1994

[9] Turing, Alan. (1936). On Computable Numbers, with an Application to the Entscheidungsproblem. _Proceedings of the London Mathematical Society_ 42 (1):230-265.

[10] Wolfram, Stephen. (1989). A New Kind of Science, Bak, Chen, and Creutz. Soc/game of life . Nature, 342:780.