Journal of Intelligent and Robotic Systems **22:** 87–95, 1998.

Book Review

Artificial Intelligence and Scientific Method, Donald Gillies, Oxford University Press, 1996, 176 pp.

Artificial Intelligence and Scientific Method is the title of the new book by Donald Gillies published in September, 1996. The book has an interdisciplinary character; it connects the fields of Artificial Intelligence (AI) and Philosophy of Science. The recent advances in AI have remarkable implications and cast new light on the field of Philosophy of Science and more particularly on the study of Scientific Method (the way science is done). The long debate about the nature of scientific method, known as the *inductivist controversy*, takes a different form after examining advances in AI, such as machine learning and logic programming.

On the other hand, it is true that philosophy has implications for AI. Many thinkers (namely Godel, Lucas, Penrose) have argued that there is a limit to what can be achieved by AI. That gives rise to questions of whether computers might become more intelligent than humans. The author gives a careful consideration of all these aspects and questions, focusing on particular examples rather than attempting a full survey. Without being absolute, he goes on to suggest a new framework for the study of logic and a viewpoint of the relationship between humans and 'thinking' computers. The whole material contained in the book serves as a challenge to the reader for further study and thinking.

The book is divided into six chapters of diverse length. Chapter 1 serves as an introduction to the inductivist controversy and the major representatives of the competing parties. Two case histories of science are presented to emphasize the similarities and the differences. Chapter 2 gives an account of the recent results of the research in the area of machine learning. Two well known learning systems (ID3 and GOLEM) are presented as well as their achievements. In Chapter 3 the consequences of these results for the controversy are explored. The author shows how a theory developed in 17th century (by Bacon) found evidence in the last few decades. In Chapter 4 the concept of logic programming is introduced and a general consideration of the foundation of logic is given. Chapter 5 integrates the concepts of the previous chapters to answer the question of whether there exist an inductive logic and if so, what form it should take. Finally, in Chapter 6 the other face is considered: the implications of philosophy for AI. The author proceeds from presenting the existing views, to criticizing them and expounding his own views about the future of AI.

The book is full of quotations and references to appropriate sources. The author makes an excellent use of the English language and the exposition of the different concepts runs fluently. Technical knowledge is not presupposed and almost everything is explained informally, but accurately. However, a general knowledge of these topics is necessary for best understanding. Some particular aspects (e.g., rlgg in relational learning, rival logics, etc.) should be more refined, for it is difficult enough for the average reader to understand them in the way they are given. Finally, some repetitions and the extensive use of first person might be annoying for some readers. What follows is an attempt to summarize the contents of each chapter and present the main idea, occasionally invoking the author’s own words.

Chapter 1. The Inductivist Controversy, or Bacon versus Popper

The *inductivist theory* of scientific method (inductivism) was developed by Sir Francis Bacon (1551–1626), basically in his famous treatment *Novum Organum* (1620). The basic idea is simple. A scientist should begin by making a large number of careful observations. Then, from this mass of data, laws should be extracted by a process known as *induction*. However, this type of induction is supposed to be a mechanical method, independent of the scientist’s intuition, known as mechanical or Baconian induction.

The theory of conjectures and refutations or *falsificationism* was developed by Sir Karl Popper (1902–1994). According to this alternative approach, the scientist starts with conjectures and then tries to refute (or falsify) them by criticism and testing (experiment and observations). In this case, the obtained conjectures are due to the creative thinking and intuition of the scientist.

A traditional method of weighing the merits of competing theories in the Philosophy of Science is to analyze case histories of what is generally acknowledged as good science, in order to see whether they fit the theories under consideration. The Kepler’s discovery of the laws of planetary motion and the discovery of the sulphonamide drugs are used for this purpose. After careful consideration, the honors are divided somewhat between Popper and Bacon. However, the long dispute between inductivists and falsificationists takes another form in the light of the recent advances in AI.

Chapter 2. Machine Learning in the Turing Tradition

An *Expert System* can be defined as a program designed to perform the task normally performed by a human expert. It consists of two parts: (1) the knowledge base, and (2) the inference engine. Expert systems were invented by the *Stanford*

Heuristic Programming Project and the first such system is known as DENDRAL (Feigenbaum, Buchanan, et al.) which was used in mass spectrometry.

The major problem in the development of expert systems is the elicitation of the knowledge from the expert(s). It was formulated in 1977 and it is known as the *Feigenbaum's Bottleneck Problem*. It is generally admitted that the problem is not only a matter of time. The problem gave rise to the development of machine learning methods as a solution.

Before proceeding a distinction should be made. There are two different approaches in the machine learning area. The first, the *psychological approach*, represented by Simon and his group, attempts to simulate the inductive rules of inferences and the thought of famous scientists, in order to rediscover (or discover new) laws using the machine. But the thought of the famous scientists is very complicated, not analyzable to simple inferences. There are no surprising results by this approach and no connection with the practice. On the other hand, the *logical approach or Turing tradition*, goes back to Turing and combines logic and practice. The object is to obtain general inductive rules, using logic and probability, in order to make original discoveries. There are many significant results in the area and applications in practice. Machine learning will be considered under the Turing tradition within this context.

Three basic characteristics of the machine learning systems are the following:

1. They use *Inductive Rules of Inference* (IRIs), but with a mechanical nature.
2. The IRIs take the form From K & e, derive H, where K is the background knowledge, e is the evidence (or data) and H is the hypothesis generated to explain e using K.
3. They apply the IRIs iteratively.

The machine learning methods can be divided between (1) Attribute-based Learning, and (2) Relational Learning. In the first case, the problem is to classify objects on the basis of a set of attributes. A training set of examples is given, where each member is fully described in terms of some attributes. There is also an explicit distinction between positive and negative instances. The object is to induce a correct classification law that distinguishes positives and negative instances using the attributes. This method is connected to Aristotelian Logic (each attribute is a 1-place predicate) and to propositional calculus (the law is built by combining attributes using the connectives & – and, V – or, $\neg$ – not and $\rightarrow$ – implies). On the other hand relational methods use first-order predicate calculus and the instances of the training set are described in terms of n-place predicates (relations). The machine learning algorithms can be top-down or bottom-up.

Quillan's ID3 is a top-down, attribute-based learning system, based on CLS (Concept Learning System). It makes use of decision trees. Each level in a decision tree corresponds to an attribute. The branches under a node at some level correspond to the different values of the attribute at this level. Finally, all the leaves correspond to either positive or negative instances. The procedure of building a decision tree runs as follows:

1. Select an attribute (this is a very important step).
2. If all the positive (or all the negative) instances belong to one branch, then you are done.
3. Otherwise, apply the same procedure to each branch which has both positive and negative instances.
4. Terminate when all the data have been divided.

CLS selects attributes using a cost function, whereas in ID3 the selection is based on the information gain. As soon as the decision tree has been built it can be transformed easily (mechanically) to a rule. The IRIs here appear in filling an incomplete level of the tree by choosing attributes that maximize information gain. The background knowledge appears at the choice of the attributes and the obtained hypothesis is that the resulting tree is a correct classifier which is tested for falsification. ID3 is employed in a top-down fashion. A decision tree is built for a subset of the data (window) and the rest of the data are classified using this tree. If the classification is correct then it terminates, otherwise the procedure is applied on a new window consisting of the old window and the incorrectly classified objects. All this method is successful but it has many limitations especially when 'noise' in the data is permitted.

GOLEM is an other learning system which uses the concepts of subsumption and *relative least general generalization* (rlgg)* in first-order predicate logic. The training set consists of positive and negative instances and a set of relations. GOLEM selects randomly some pairs of positive instances and constructs the rlgg for each pair. Then it computes the positive and negative instances that can be predicted by these rlggs and chooses the one, say S, which predicts the most positive, while the number of negative is under a predefined threshold. Then the same procedure is applied on S along with some more random pairs. The procedure is bottom-up and the IRIs appear in the formation of the rlgg of a subset of the data. Then, falsification eliminates the 'bad' rlggs.

Chapter 3. How Advances in Machine Learning Affect the Inductivist Controversy

Practice (experiments and experience) has shown that expert systems have a better performance than the human specialists. Moreover, when machine learning techniques are used, instead of the conversation method, for the knowledge acquisition, the results are better. It is worth to note that GOLEM has discovered a law of nature hitherto unknown (it is simple law, but the important is that it has been discovered by a machine). All these advances have established a human-computer interaction which gives rise to research at new directions.

* The definitions are omitted here, because of their complexity.

The Baconian method had not been used up to the present years because of the absence of physical instrument for mechanization of thought. But now, the computer industry has provided the physical instrument for the application of Bacon's method. Bacon's example of heat comes is correspondence with machine learning systems. However, Bacon had not mentioned the need of background knowledge used in AI, which gives credit to Popper. In any case, AI has changed the way science is done.

Chapter 4. Logic Programming and a New Framework for Logic

PROLOG arose when two lines of investigation were brought together: Automatic Theorem-Proving and Natural Language Processing. An important characteristic of PROLOG is the *negation-as-failure* employed instead of the classical negation. Negation-as-failure means that given a predicate its negation is considered true if the system fails to prove the predicate. As a result, the logic in PROLOG is a *non-monotonic* logic. In classical logic, if a conclusion follows from some premisses, it still follows however many additional premisses we add to the original set. The number of conclusions can only be increased by adding premisses. Thus, the term monotonic logic. But with the negation-as-failure it is not any more true. Moreover, PROLOG with negation-as-failure is not equivalent to classical logic with the closed world assumption (i.e., everything not explicitly given is considered false).

Several translations of non-classical logics to classical logic have been proposed but this does not mean that they are in fact classical logic in disguise. This is strengthened by the Godel's translations of classical logic into intuitionistic logic. A translation from logic L to logic M is *formally correct*, if it maps any theorem of L into a theorem of M. An *adequate translation* from logic L to logic M is a formally correct translation which preserves the meaning of the key terms. Thus, because of the different meaning of negation, every translation of PROLOG into classical logic will be formally correct but not adequate.

A more profound difference is that PROLOG introduces control into logic. Kowalski, in his ideas of logic programming gives the following equation: Algorithm = Logic + Control. The logic program is the logic component of the algorithm, whereas the control component is the program executor. An extension of this equation can be formulated as follows: Logic = Inference + Control. The inference component is the set of rules of inference for derivation of conclusions, whereas the control component is the set of meta-rules used as a guidance to which assumption or rules of inference to choose. The introduction of control into deductive logic by PROLOG can be strengthened by an analogy: the replacement of craft skill by mechanization in mathematical proofs.

Classical logic (and intuitionistic) helps the mechanization of the process of checking the validity of a proof. PROLOG involves not only this, but, also, the mechanization of the construction of proofs. In order to construct proofs, PROLOG searches systematically the various possibilities in a depth-first, left-to-right fashion with backtracking. Each clause in PROLOG has two interpretations: a declarative and a procedural. For example, the clause 'white(x):– swan(x)' has a declarative interpretation that 'all swans are white' and a procedural that 'in order to show that x is white you must show that x is a swan'.

An other important feature of PROLOG are the control elements ! and fail. ! prevents backtracking and fail causes the interpreter to fail. The negation-as-failure can be defined using ! and fail as follows:

not x:– x, !, fail

not x.

Thus, PROLOG with negation-as-failure can be seen as a symptom of the more profound difference that PROLOG introduces control into deductive logic. An other difference is that the order of premisses does not make any sense in classical logic, whereas in PROLOG is important. In general, PROLOG should not be seen only as a programming language, but as a system of logic.

The concept of *certainty* in logic is that if the premisses were certainly true, the conclusion had to be true as well. Classical logic preserves certainty, but this is not the case in PROLOG. This is due to the negation-as-failure and is connected to the subject-matter with which the logics are designed to deal. Thus, classical logic is suitable for formal number theory, whereas PROLOG is suitable for timetable or similar problems, i.e., problems which involve some uncertainty.

As a conclusion, there is not a single universal logic, but different logics may be appropriate in different problem situations. Classical logic appears to has a universality (in fact it does not have), because it underlies a body of mathematics which has an enormous number of practical applications.

The philosophical thesis of whether logic is empirical or a priori, in the light of PROLOG, lead to the conclusion that logic is indeed empirical than *a priori*. The a priori conception of logic claims that the truths of logic are known a priori independently of experience, whereas the *empiricist* claims that the truths of logic are established by experience of their application.

Chapter 5. Can there be an Inductive Logic?

An old question in philosophy of science asks whether there can be an inductive logic or not. Traditionally, it was assumed that logic has two branches: deductive and inductive. This distinction led to a divergence between them. Due to the attempts to reduce inductive logic to confirmation theory, these two logics appeared to be very different things at around 1950–1960. But, the recent advances in AI have reversed the divergence. Machine learning methods introduce the inductive rules of inference and PROLOG introduces control into deductive logic. These things can be put together within a common framework. That leads to a rapprochement of inductive and deductive logic.

The difference between inductive and deductive logic can be understood by an example.

Deductive Logic:

All men are mortal.
Socrates is a man.
Therefore, Socrates is mortal.

Inductive Logic:

Socrates is a man, and Socrates is mortal.
Plato is a man, and Plato is mortal.
Einstein is a man, and Einstein is mortal.
....Therefore, all men are mortal.

Deductive logic uses rules of inference, such as modus ponens, together with axioms. The elimination of axioms in favor of rules leads to natural deduction. On the other hand *inductive logic*, proceeds from an indefinite mass of particulars to a general conclusion or hypothesis. What is crucial here is the probability of the hypothesis h given evidence e, $P(h, e)$ and the degree of confirmation which evidence e gives to hypothesis h, $C(h, e)$.

The general view was that inductive logic cannot be mechanized (formulated). But, now they can both be combined under the equation: Logic $=$ Inference $+$ Control.

Chapter 6. Do Godel's Incompleteness Theorems Place a Limit on AI?

The study of logic and scientific method has, also, implications for AI. The focus will be on Godel's Incompleteness Theorems. Some thinkers have argued that these theorems pose a limit to what can be achieved by AI.

The success of AI has raised anxieties and questions concerning the superiority of humans to computers and the possibility of a role reversal. It is a fact that nowadays there exist computers able to perform tasks that had previously been the exclusive province of human thinking. As an example, the latest chess players are not only play 'good' chess but they are among the few 300 human experts in chess worldwide. Although people welcome all the machinery that help us in a natural way (cars, airplanes, calculators), a hypothetical machine able to 'think' makes them circumspect; the human superiority is due to this ability of thinking.

This is not, however, the whole story. There is a number of people who feel pleased at the thought of a thinking machine. Among them is Alan Turing who proposed in 1950 the famous Turing Test to test the ability of a machine to simulate human thought. Turing seems to believe that although the human brain may be not a Turing Machine, its output can none the less be simulated by a Turing Machine. Those who hold the opposite view (namely Lucas, Penrose and Godel) used the Godel's incompleteness theorems to refute Turing's opinion.

Godel's incompleteness theorems refer to formal systems for arithmetic. The first of them can be stated (informally) as follows:

> Given any formal system S, such that (1) S is consistent and (2) a sufficiently large amount of arithmetic can be derived in S, then we can find an undecidable proposition p in S – that is to say, a proposition p such that p and the negation of p (not-p) cannot be proved in S; p can, however, be shown to be a true statement of arithmetic by an informal argument outside S.

The theorem can be established by giving a statement p not provable in S and indeed roughly speaking it has the form:

> This formula is not provable in S.

In order to produce such a formula, since S is a formal system of arithmetic, Godel used a process of assigning numbers to statements in S, known as Godel numbering. This theorem refutes the logistic philosophy of mathematics, given basically by Russell and Whitehead in their *Principia Mathematica*. The second Godel's incompleteness theorem runs (informally) as follows:

> If S is consistent, then the consistency of S cannot be proved within S itself.

This theorem refutes the formalist view of mathematics, held by Hilbert and known as *Hilbert's programme*.

In 1961, Lucas used the first incompleteness theorem to show that mechanization of thought is impossible. The Lucas argument states that 'however complicated a machine we construct, it will, if it is a machine, correspond to a formal system, which in turn will be liable to the Godel procedure for finding a formula unprovable-in-that-system. This formula the machine will be unable to produce as being true, although a mind can see that it is true. And so the machine will still not be an adequate model of the mind'. He proposed also a 'game' between himself and a mechanist where the mechanist proposes a model of mind, Lucas points out something that this model cannot do, but mind can, the mechanist changes the model and proposes it again to Lucas and so on until one of them cannot continue.

An objection to Lucas's argument, coming by Putnam, states that Lucas assumes that the system S is consistent. But he is no more capable of proving this than the machine. This does not give an answer because, although difficult to give a full formal proof for the consistence of S, human can give intuitive, informal reasons to believe this, whereas the machines lack this ability. Lucas's object was to show that human mind is different than a machine, but he regards the mind as superior to the machine. Lucas argument seems to support dualism. Since mind can always go better than a material artifact, it is natural to suppose that it has a different character or substance from matter.

On the other hand, Godel's opinion was close to Lucas's although not the same and advocated both dualism and the superiority of human mind. It runs as follows: ‘Either the human mind surpasses all machines (to be more precise: it can decide more number theoretical questions than any machine) or else there exist number theoretical questions undecidable for the human mind’. However, it contains a stronger premiss: the human mind can decide all number theoretical questions.

An other thinker, Penrose, goes on to argue that any properly intelligent entity must be conscious, and so, from his claim that consciousness involves non-algorithmic processes, could not be a computer. He suggests, also, that a new theory of physics is necessary to capture the human mind processes.

There are numerous objections against Lucas’s, Penrose’s and Godel’s arguments. For example, Benacerraf considers the possibility that we are all Turing machines, but each of us is unable to get to know his or her machine table or program. This argument has been refined by Chihara and indeed applies well against the opposite arguments considered so far.

Many objections come out when considering the question in the light of the recent advances in AI. Godel’s incompleteness theorem applies to static formal systems but human mind cannot be represented by such a system, because of its dynamic nature. Moreover Lucas argument suffers by the representability assumption, i.e., that any machine can be represented by a formal system.

As a conclusion, there is no satisfactory argument, based on Godel’s theorems to establish the superiority of minds to computers or the reverse. However, the mind will always remain superior to computers in a somehow different sense: *Political Superiority*. Computers are built to carry out human purposes. If a computer fails to do so, it can be switched off by humans. Thus, mind is politically superior to machines. A second conclusion, is that the advances in computing and AI will establish a new human-computer interaction, which is more likely to stimulate human thinking than to render it superfluous.

Artificial Intelligence and Scientific Method is a good book that stimulates the human thought to look at the past, the presence and the future of science in a critical manner. The ‘audience’ of the book can be anyone who is interested in aspects concerning science and philosophy and especially the scientists who believe that their title (PhD – Doctor of Philosophy), apart from a narrow area of expertise, requires a broader overview of what is considered to be science today.

MICHAIL G. LAGOUDAKIS

The Center for Advanced Computer Studies,
University of Southwestern Louisiana,
P.O. Box 44330, Lafayette, LA 70504
e-mail: mgl4822@usl.edu