Assisting Scientific Discovery with an Adaptive Problem Solver

Christopher Dartnell[1] and Jean Sallantin[2]

[1] EURIWARE, Octeville
44 Rue des Vindits, 50130 Cherbourg-Octeville - France
christopher.dartnell@euriware.fr,
[2] LIRMM, UMR 5506
161 rue Ada, 34392 Montpellier Cedex 5 - France
js@lirmm.fr

Abstract. This paper is an attempt to design an interaction protocol for a multi-agent learning platform to assist a human community in their task of scientific discovery. Designing tools to assist Scientific Discovery offers a challenging problematic, since the problems studied by scientists are not yet solved, and valid models are not yet available. It is therefore impossible to create a problem solver to simulate a given phenomenon and explain or predict facts. We propose to assist scientists with learning machines considered as adaptive problem solvers, to build interactively a consistent model suited for reasoning, simulating, predicting, and explaining facts.

The interaction protocol presented in this paper is based on Angluin's "Learning from Different Teachers" [1] and we extend the original protocol to make it operational to assist scientists solve open problems. The main problem we deal with is that this learning model supposes the existence of teachers having previously solved the problem. These teachers are able to answer the learner's queries whereas this is not the case in the context of Scientific Discovery in which it is only possible to refute a model by finding experimental processes revealing contradictions. Our first contribution is to directly use Angluin's interaction protocol to let a machine learn a program that approximates the theory of a scientist, and to help him improve this theory. Our second contribution is to attenuate Angluin's protocol to take into account a social cognition level during which multiple scientists interact with each other by the means of publications and refutations of rival theories. The program learned by the machine can be included in a publication to avoid false refutations coming from a wrong interpretation of the theory.

1 Introduction

Assistance to Scientific Discovery is a very challenging research domain: scientists study open problems which have never been solved. Therefore, no satisfying model or theory might already exist, so we propose to assist scientists by learning machines in their task of theory building. Such a software assistant can learn to simulate an observed phenomenon explain or predict facts, has to deal with uncertainty, pattern discovery, interactive ontology building [2], and has to produce statements comprehensible to a human to improve human-machine interaction. We divide the problem solving process in 3 steps:

1. The user, acting as the teacher, interacts with his learning assistant to make him learn his hypothesis. To model the interaction between a scientist validating his hypothesis with his assistant, we directly exploit Dana Angluin's Learning from different teachers paradigm [1] which formalizes a protocol for a human compliant robust learning defined as the result of a stable interaction cycle between a Learner and a Teacher. Her conclusions are completed by theoretical results about query driven machine learning complexity in [3] and [4].
2. The assistant brings a critical attitude concerning an approximation of the user's hypothesis to confirm or invalidate his hypothesis, and this interaction can lead to a revision of the hypothesis and/or of the description model, in which case they are both considered as learners. The user, at the heart of the system, builds interactively with an adaptive problem solver an adequate description model of the studied phenomena: he is in charge of providing a description model, and the adaptive problem solver uses machine learning and paraconsistent logic to detect contradictions between the learned theories and empirical results, or inadequateness between the description model and learned theories. These contradictions are used to initiate an Angluin-like interaction cycle during which the user learns at the same time as the machine, and this co-learning leads to a pertinent understanding of the problem.
3. Once the user considers that the approximation of his theory learned by his assistant is expressive enough, he can use it to publish his own theory: each theory proposed by a scientist is not refutable, but a logical theory produced by a machine can always be reduced to a universal form which is refutable by an existential statement. Our contribution is to extend Angluin's protocol by introducing a social interaction level inspired by Popper's philosophy of science [5] and based on proofs and refutations of publications. The publications are logical conjectures which have to be submitted to the judgment of other learners to be pitilessly tested, put into question, and eventually falsified.

In section 2, we define the needed functionalities that a problem solver should implement to be adaptive and autonomous, and we emphasize that such an adaptive problem solver has to reason in paraconsistent logic to cope with contradictions. We show in section 3 that these contradictions are at the source of the interaction between the solver and the human it's assisting, and how this interaction is formalized in [1] by the use of *Membership and equivalence queries*. However, this learning model supposes an access to a Teacher to answer these queries whereas there isn't any to help scientists understand Nature and its laws, so we propose in sections 3.2 and 3.3 two extensions of this model to make it operational in the context of scientific discovery, and we validate in section 4 this protocol on a toy game, E+N. Finally, we present some experimental results before concluding.

2 Toward a Definition of an Adaptive Problem Solver

Common definitions of a problem solver take into account the type of solvable problem which characterizes it, as a differential equation problem solver, or nonlinear equation systems solver: a problem solver is designed to perform the computation of a known problem which has already been solved and modelled. So for any presented instantiation of the specific problem, it is able to solve it and produce its solutions.

An adaptive and autonomous problem solver should be able to acquire new abilities by learning how to solve new problems, and use this knowledge and experience to find solutions. To solve an open problem, one has to observe the problematic situation and analyse it to build a language describing the situation's dimensions pertinent for reasoning. These dimensions determine the definition domain of the variables characterising the problem and influencing the solution's computation. The language thus defined is used to formulate assumptions and hypothesis that have to be experimented. Comparing empirical results and theoretical computations can reveal contradictions between a theory and reality, and therefore lead to a revision of the description model and to the formulation of new hypothesis.

By making an analogy with the process of scientific discovery, in which neither the *ontology* nor the *theory* are perfectly known *a-priori*, we define below the functionalities that an adaptive autonomous problem solver should be empowered with to assist the process of discovery. It should be able to build and maintain an *Ontology* of the domain. By *Ontology*, we mean a logical language relevant with observations describing the pertinent dimensions of the problem, i.e. the types of the variables involved in its resolution. Furthermore, we want the ontology building to be the consequence of the interactive learning process of the logical language.

The principles of nominalization and reducibility [6] are the keys of a problem solver's adaptability, since they allow it to manipulate new concepts and design experimentations to validate the pertinence of these new dimensions for the computation of the problem's solutions:

- The solver should be able to learn ontological statements to constraint the relations between the values of the problem's dimensions, by analysing and correlating gathered information.
- The solver should be able to theorize: discover, name, and symbolically use regularities or patterns appearing on data by revising the ontology and introducing new dimensions to the problem's formulation. Transforming an observed property into a symbolic object and re-use it is called the *Nominalization* principle. This principle is essential to formulate and express a *theory* to explain the problem and predict further results. By *theory*, we mean a set of rules used to compute a problem's solutions.

– The solver should be able to empirically validate theories: transcribing mathematical abstractions to design experimentations feasible in the real world is called the *Reducibility* principle.

Interactions between the solver and its environment are *sine qua non* conditions of its evolution: by comparing the results of theoretical computations and the results of its interactions with the environment, the solver is able to detect contradictions in the formulated theories. These contradictions are used to motivate the actions and reflections of the adaptive problem solver: each experimentation is made to validate a theory, and is preceded by a prediction about its consequences. This prediction is compared to observed results to search for contradictions. Of course, the most informing situation is when a contradiction is detected, because it reveals either a wrong formulation of the problem by the user (perhaps a parameter was forgotten), or a inconsistency in the learned theory (coming from a bias in the learning set). To reason in the presence of contradictions, the logical ontology must be paraconsistent [7]: paraconsistent logics don't allow absurd reasoning (*ex-contradiction sequitur quod libet*), i.e. a statement and its negation can be true at the same time.

The following deduction shows that the paraconsistent contradiction principle requires four arguments to deduce a contradiction about A:

$$\frac{\neg A \vdash B \quad \neg A \vdash \neg B \quad \neg A \quad \vdash \neg(B \wedge \neg B)}{A \wedge \neg A}$$

$$\frac{\neg pope \vdash rain \quad \neg pope \vdash \neg rain \quad \neg pope \quad \vdash \neg(rain \wedge \neg rain)}{pope \wedge \neg pope}$$

In this example, all the arguments are evaluated. A contradiction Only if the contradiction "$\vdash \neg(rain \wedge \neg rain)$" is not admitted, then "it is paradoxical to be pope".

This is very useful when reasoning on descriptions coming from different contexts or points of view, and [8] gives an elegant example of paraconsistency based on a defeasible deontic logic:

– a *Paraconsistent Logic* allows to reason in presence of contradictions in order to maintain obligations.
– a *Deontic Logic* allows to maintain a past knowledge by using the *Obligation*, *Forbiddance* and *Permission* modalities. In the context of machine learning, an obligatory fact is a fact which provoques a major contradiction when false, a forbiden fact is a fact which provoque a major contradiction when true. An advised (not obligatory) fact will provoque a minor contradiction when false, a disadvised fact (not forbiden) will provoque a minor contradiction when true.
– a *Defeasible Logic* allows to revise the model when new contradictory facts occur, and to produce a new theory adapting the strength of the contradictions.

Deontic logic is used to localise contradictions and provoke a revision in the set of defeasible theories, and this paraconsistency allows the solver to adapt the ontology to new facts and new observations.

Paraconsistent defeasible deontic logic rules describing a complex system are not easy to determinate, and since this kind of monotonous and multivaluated logics have experimentaly been shown as learnable using Angluin's paradigm [2], our goal is to have them learned by an adaptive problem solver interacting with a human.

In the following section, we discuss how Angluin's interaction protocol for machine learning can be used to formalise the necessary interactions between such a solver and its environment. We use contradictions to drive this interaction. We also propose an extension of this protocol to adapt concept learning theory to scientific discovery.

3 Making Angluin's Formalism Operational in the Context of Scientific Discovery

Angluin's formalism [1] gives a strong basis to interactive learning from different teachers, and introduces the idea that a learner could possibly become a teacher for another learner. We present this formalism in section 3.1, then we apply this protocol to assist scientific discovery with a learning machine in section 3.2. We show in section 3.3 how we introduce a social interaction level between learners to make this protocol operational in the context of scientific discovery, *i.e.* to cope with the apparent impossibility to use Equivalence queries.

3.1 Formal Aspects

Formal learning models differ by the information sources, by *a priori* knowledge given to the Learner, by its tasks and abilities, and by the success criteria of the learning process. In the model of exact identification with queries, studied in [3] and [4], the task is to identify an unknown concept drawn from a known concept class using queries to gather information about the unknown concept.

The interest of Angluin's works lies in the theoretic results she provides about the learnability of different concept classes (as monotonous DNF which are not learnable in the case of PAC learning or online learning) by methods based on the use and the combination of two main types of queries: *Membership* and *Equivalence* queries defined as follows:

Let the *domain* X be a nonempty finite set. A *concept* c is a subset of X, and a *concept class* C is any nonempty set of concepts. In a *Membership query* (MQ), the learner exhibits an example $x \in X$, and the access to an oracle returns 1 if $x \in c$, and 0 if $x \notin c$. In an *Equivalence query* (EQ), the learner exhibits a concept $c' \subseteq X$, and the oracle's answer is either "yes" if $c' = c$, or an element x in the symmetric difference of c and c', if $c' \neq c$. In [9] and [4], Angluin demonstrates the necessity of combining MQs and EQs to allow a powerful and effective learning. [1] formalizes a learning model based on the interaction between a *Learner* L and a *Teacher* T. Both of them are modelled

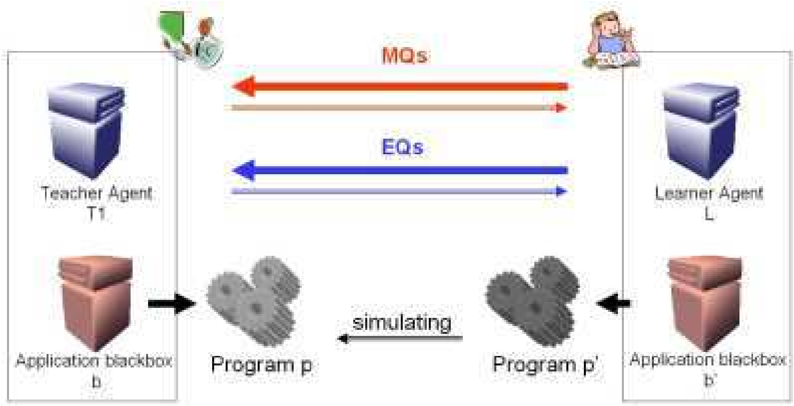

Fig. 1. Exact identification with queries

as computers, and T is assumed to have a program p representing the concept to be taught to the learner, as illustrated in figure 1.

The teaching protocol involves examples of the concept, and possibly *other information* (we bring up this topic again later in this section) from which the learner is to develop a program p' that also represents the target concept. [3] emphasizes the fact that outright coding, in which T would transmit (using an encoding via examples) the text of the program p to L, is neither compliant, nor representative of human learning. Indeed, the "hardware and software environment" differs quite substantially from one person to another. In other words, all of us don't have the same brains, nor the same ways of thinking, although our anatomies are comparable.

Angluin illustrates this point of view with the human learning of juggling, which brings in muscular and visual reflexes, time perception and so on... These are "low level" mechanisms from which we only know very few about their triggering and their control. We merely know how to use or how to interpret their inputs and outputs, doing so in a symbolic way. The idea we want to put forward is that *to realize a task, acquire an ability, or identify a concept, the learner has to learn how to correctly use and combine "black boxes" representing the mechanisms triggered during the execution of the task, which are only partly and poorly known to him.* This example gave us the motivation to test this protocol in the context of scientific discovery (see section 3.3): according to Popper's conception of Scientific Discovery [5], scientists try to understand Nature's laws, by designing experiments and formulating theories on the basis of their results.

We show in section 3.2 and 3.3 how Angluin's protocol, in such a context, can be used at different levels of interaction.

Here is the theorem enunciated in [1]: *There exists a learner L^* such that for every total recursive function b(x, s) there is a teacher $T * b$ such that for every universal function box g' and every function box g that is b-related to g', L^*(g) learns all the partial recursive functions from $T * b$ (g'). Furthermore, L^**

is box-and-teacher-proof. Using works as [10], [1] demonstrates the identification in the limit [11] of this process.
This means that *whatever might T's and L's applications black box be (applications being formalized by Angluin by recursive functions), L will learn after a finite number of queries, a program p' simulating p and producing only a finite number of errors if:*

1. *The computing performances of T and L are comparable, which means L is "not too slow" compared to T,*
2. *T has already managed to solve the problem.*

This theorem stands in the context of language learning, to which the context of scientific discovery is comparable since scientists aim at learning or discovering a language adapted to describe their environment and various phenomena occurring in it.

This protocol is clear and simple, and it ensures the convergence of the learning process, or at least, it ensures that whoever might the teachers be, the learner will not converge towards an incorrect solution. An adaptive problem solver can then learn a theory formulated by a scientist during the process of scientific discovery. However, in the context of scientific discovery, the Nature, which is considered as the teacher, is "silent" and cannot answer all learner's queries: the learner may still use MQs, by designing experiments and interpreting their results, but there is no way he can access Nature to answer his EQs ("is earth flat?", "is the law of gravitation true?").

We show in section 3.2 how the learning assistant can bring a critical point of view to the scientist while analysing experimentations' results and formulating theories, and in section 3.3 how we extend the model "Learning from Different Teachers" to "Learning from each other" by introducing multiple learners and interactions between them to confront a learner's interpretation of Nature with others'.

3.2 Interactive aspects: Individual reasoning

A scientist L learns from Nature T, by experimenting his hypothesis. In our approach of assistance to scientific discovery, we want the scientist to interact with an adaptive problem solver to find the solution of a problem, and the solution comes from the co-learning of these two entities.

An intelligent assistant is an adaptive problem solver, as described in section 2, able to analyse facts described in the language of the ontology written by the researcher while observing and describing the problem. The intelligent assistant that we develop [12] uses induction and abduction methods coming from machine learning with graphs and Galois lattice theory [13] which allow to find relevant logical implications and equivalence rules between the descriptors introduced by the user to describe the facts observed (see fig. 2). These rules can be easily understood by the researcher since they are formulated with his own words.

The assistant can then induce theories predicting the behaviour of the studied system, and use abduction to explain past facts and to design experimentations testing the validity of the produced logical rules.

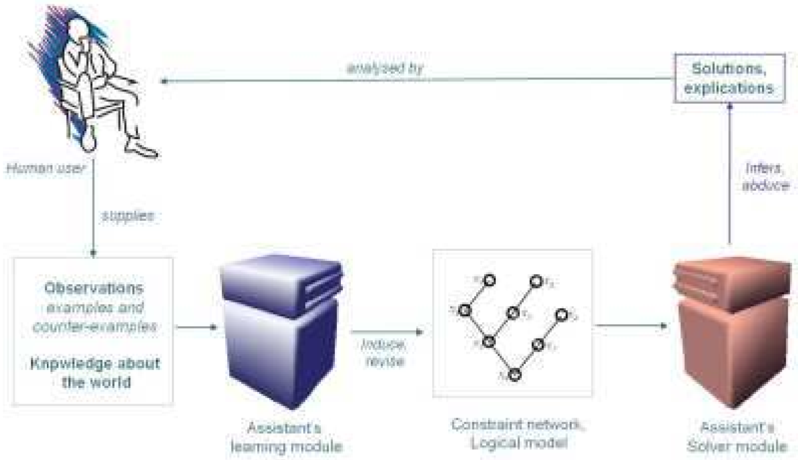

Fig. 2. Human-Machine interaction cycle

Compared to Angluin's protocol, we propose to let the Learner be a couple of Learners: a scientist and his assistant, learning as well from their common Teacher (Nature) than from each other. The interaction between them follows the protocol presented in section 3.1, with this difference that both entities can act in turn either as the Teacher or as the Learner. So the scientist is in charge of formulating a description model of the problem's domain, and to modify it when irrelevant examples arise: erroneous predictions invalidate either the theories, either the initial conditions (description model). Therefore, he represents the teacher guiding the learning machine, and can also eliminate learning errors coming from bias of the learning set, for example by designing experiments to produce results considered by him as informant. The assistant can analyse large data and formulate an opinion concerning the scientist's choices, and act as the Teacher by anticipating negative result of MQs or EQs.

This justifies the use of a paraconsistent defeasible deontic logic to localise contradictions in the scientist's interpretation of results, in the approximation of his theory learned by the assistant, or in the discretization of the problem by the scientist, *i.e.* in his description model. We saw how Angluin's protocol formalized the interaction between a scientist and his assistant, and how the link to their common Teacher was made by designing experimentations and interpreting their results. The next section deals with the need of a social game between learners to answer one's EQs.

3.3 Social aspects: Collective cognition

As we introduced in section 3.1, the Nature, which is considered as the Teacher from which scientists learn during the process of scientific discovery, cannot be accessed to answer EQs, so we introduced a social interaction level to answer these queries. A scientist is member of a community, and published theories are temporary solutions accepted until they become insufficient to explain Nature: in our model, learners (who are couples of scientists and their assistant) are confronted to the judgment of other learners to cope with the impossibility to access an oracle for EQs in the context of scientific discovery. Every learner has the same access to Nature for MQs, whether they have proper interpretations and points of view, and they are in charge of answering other's EQs, as shown on fig.3.

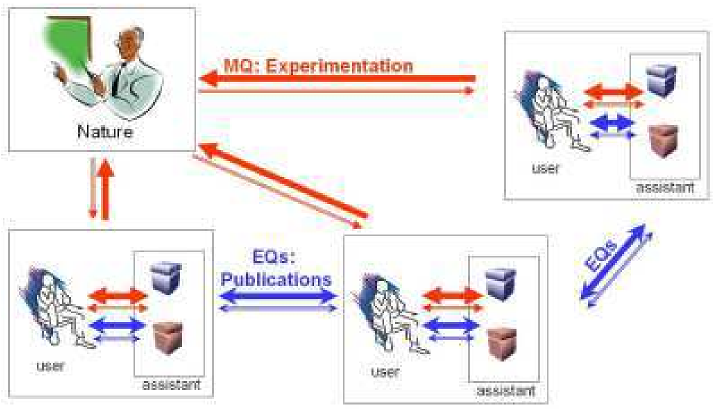

Fig. 3. Coping with EQs without an oracle

Angluin's prospect of letting a learner becoming a teacher is meaningful for us, and our model let other learners answer EQs by the means of publications and refutations. Doing so, we allow the learners to act on behalf of a teacher by refuting other's hypothesis. According to "Learning From Different Teachers" theory, learning is still possible in these conditions (if the learners are *teacher-proof*). We symbolize the product of this social interaction by a score and a profit function. By attributing or deducing points for each query, depending on the oracle's answer, we can create a competition atmosphere or collaborative work between multiple learners. This atmosphere motivates the emission of EQs to score points and MQs to prove or refute a theory. The introduction of this social level can lead to experiment different points attribution in order to determine in which condition the community formed by the learners converges faster to an acceptable solution. These kinds of experiments are planned by cognitive scientists [14], and some of them have already taken place with human players only.

We shall now describe these experimentations with the game *E+N* and link the manipulated concepts with the notions presented previously.

4 Protocol validation on a toy game: $E + N$

We implemented this toy game E+N to validate the protocol of assistance to scientific discovery presented in previous sections. In this experimentation, we aim at defining the limits of our protocol, and having a standard reference to evaluate further experiments implying scientists and their assistants.

4.1 problem's definition: Eleusis

The problem in Abbott's Eleusis card game [15] is to find a secret law hidden from the players and determining the valid sequences of cards that can be played during the game. The difficulty of the game can be adapted by:

- changing the length of the sequences concerned by the secret rule, to increase the complexity of the learning problem.
- fixing the choices offered by the rule, determining the ramifications in the resolution space. This might lead to formulations of various classes of Boolean formulas, as *CNF, DNF, k-term-DNF, ...*
- giving or hiding this information to players. This allows letting the learner fix his own learning biases or not.
- providing or not the Ontology used to explain the rule. This might be equivalent to concept learning on a finite or infinite domain

Players can formulate membership queries (MQs) by proposing a sequence of cards which is accepted or rejected by an oracle machine simulating Nature, and build on the basis of their experiments a theory consistent with their current knowledge to explain the hidden rule and predict further sequences. Since a concept learning problem can be assimilated to the problem of learning the mapping function between a set of examples ($x \in X$, X being a non empty finite set) and the Boolean value representing the belonging of x to the unknown concept $c \subseteq X$, we assume that it is suited to apply concept learning theory and use the interaction protocol formalized by [1].

Experimentations are in fact membership queries ("is $g(x)$ true?", g being an hypothesis, is an MQ), with this difference that experimentations often have a cost (time, resource, ...). [1] showed that algorithms using only membership queries were less performing than algorithms using membership queries combined with equivalence queries ("does $g = f$?", f being the hidden function). The problem in the case of scientific discovery is that if experimentation results can be analyzed and interpreted to estimate the answer of a membership query, it is impossible to access an oracle able to answer an equivalence query ("is earth flat?"). By taking this point into account, and to improve the rule discovery process, we introduce a social interaction level by letting learner agents join a community respecting a multi-agent publication protocol to dispatch equivalence queries to other members of the community, as described in next section.

4.2 A social interaction level to cope with Equivalence Queries: Eleusis + Nobel

We designed the card game E+N to simulate a situation of collective problem solving implementing MQs. To simulate a real problem of scientific discovery, the oracle cannot be accessed to answer EQs. In fact, it is often hard and time consuming to determine the equivalence of two elaborated theories, which might not even use the same ontology, since each researcher has a personal way of describing the world and interpreting the experiment results.

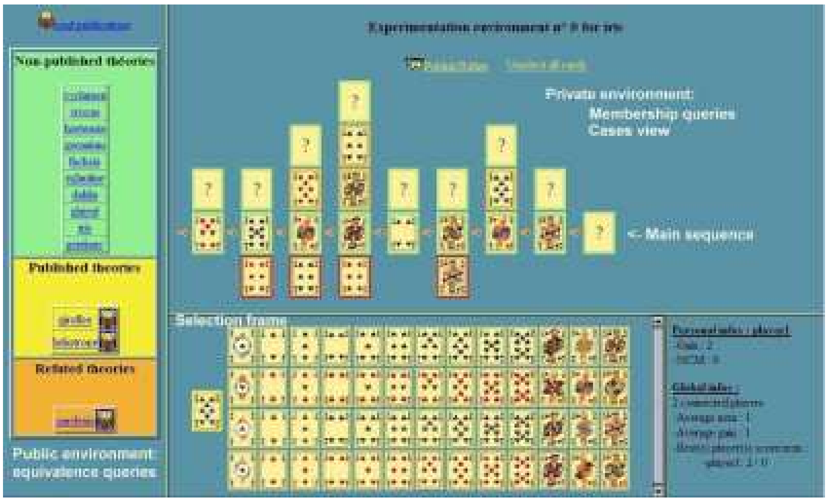

Fig. 4. Eleusis + Nobel Game display

Figure 4 shows a possible interface for E+N. This is the first we developed, as a web application, sufficient for the basic experimentations we made with human players only.

Private environment: The central frame displays the results of the experiments made by a player, by selecting cards in the bottom frame and placing them on the "? holes" to form a new sequence and submit it to the studied Nature's law. This way of displaying the results comes from Abbot's original game:

- A red surrounded card means it can not be placed after the previous one in the depicted main sequence.
- A green surrounded card means it forms a valid sequence with the previous card AND with the following one in the main sequence.
- When the card forms a valid sequence with the previous one in the main sequence BUT NOT with the following one, it is surrounded in orange.

Public environment: The frame on the left hand side displays a set of hidden rules, publicly accessible by an imaginary name so their meaning is hidden. A player can select the one he (or she) wants to study, and can switch between them whenever he wants to. It opens in the central frame a private experimentation context associated with the selected rule. When an EQ is formulated, by publishing a theory, the rule appears in the "published theories" cell, every player can read it, and the publisher scores P points. The theory is considered as correct until a player finds a counter example to refute it. The theory then moves to the "refuted theory" cell, the refuter wins R points, and the original publisher looses R points.

The P/R ratio can be set to modify the experimentation's conditions, and can also be different in the competitive scientific community and in a collaborative working group, as proposed in section 3.3.

The alternative to Equivalence queries is a publication, a broadcast of the player's theory to every other player belonging to the community. Each player can then compare this published theory with his personal current data, try to prove it's inconsistency, and refute it if a counter example is found. In fact, the theory is not broadcasted to everyone, but is added to a publicly accessible database storing every publication that is made on a hidden rule, and a notification is sent to players. This public database is a kind of collective memory, which efficiency as been shown in works on Case Based Reasoning as [16] or [17].

5 Results and comments

We present in this section the results concerning the experimentation we made to validate our protocol. To reach this goal, we needed players, an experimentation which duration can be controlled, so the rules were defined to concern sequences of only two cards, in order to have a degree of difficulty suited to non assisted human players. The results showed that a human playing alone (i.e. using MQs alone) takes between 5 and 15 minutes to publish a theory concerning a rule implying only sequences of two cards. He usually considers his theory correct, and doesn't try to refute it. Moreover, the average number of published theories is between 10 and 20 (players stop before trying the 33 rules), and few of them are equivalent to the corresponding hidden rule.

An interesting alternative was to organise duels, between two players working on the same rule, until one of them admitted, without being sure, that the adverse theory was true. The players reached a consensus on a common Ontology.

To contrast with previous results, we made further experimentations involving multiple players, students coming from different scholar backgrounds. The average time for a publication is the same, and we observe a period of roughly half an hour during which players publish. Then they begin refuting each other, and theories are revised and republished. A community of ten to thirteen players takes between 1h 1/2 and 2h two reach a stable equilibrium of published theories (opposed to the theoretical length of 5h 1/2 for one-player games). The amount of correct theories is also much more superior. This empirically validate the need to use both Membership and Equivalence queries [1], and the use of a collective memory to share experience and points of view on a given problematic.

Some of these experimentations of the protocol failed because players made false refutations caused either by a misunderstanding of the ontology or of the published theory, or even of the notion of refutation. This shows the need, even for such small learning problems, to include in the publication a program simulating the user's theory and allowing to detect eventual contradictions within it (see section 2). A second notice is that the bias coming from the P/R ratio favoured the players who only refuted others' publications without publishing themselves. We need to define an other ratio taking into account Popper's idea that a falsifying experimentation shows a contradiction in a theory, but doesn't stand without a rival theory. The third notice is that this game is very efficient to teach the epistemological foundations of science theory.

Since the protocol is validated, we will now introduce the interaction between scientists and assisting machines, but this implies more time and more efforts from the players to learn how to work with adaptive problem solvers. This has to be worth it, so next experimentations will have to last longer, for example one month. These experimentations have excited some biologists who plan to help us designing another version of the game in which the hidden laws will be real scientific discoveries as described in [18], to simulate the (re)discovery of Nobel prizes... We will organize very soon an experimentation in which a team of human players will be opposed to a team of human players assisted by intelligent learning assistants to validate our approach. An obvious use of intelligent assistants is to let them test a sequence on various published theories, and to be the guaranteeing one of the user's published theory to answer MQs. The violation of the theory is located by the contradiction between the assistant's answer and what is really observed.

6 Conclusion

We presented an interaction model to assist scientists with adaptive problem solvers in their task of scientific discovery.

[1] formalized an interaction protocol for machine learning based on the use and combination of Membership queries and Equivalence queries, that enables a machine to learn a user's theory. In the context of scientific discovery, the user is fallible, and we emphasized that reasoning in a paraconsistent logic allows the solver to localize contradictions in the user's theory, which leads to a revision of the description model. Defeasible logic is useful to supervise the learning process and "$forget$" wrong theories. In our model, the user and the software assistant act in turns as the teacher or as the learner, and this interactive co-learning leads to a better understanding of the problem and to the creation of an adequate description model; being assisted by a learning machine trivializes some fairly easy problems.

Membership Queries can be simulated by designing experiments and interpreting their results, experimentation putting the hypothesis to test. To simulate the oracle to answer *Equivalence Queries*, we introduced a social cognition level to let multiple learners interact by answering each other's EQs. Stating that a group will solve a problem faster than an individual, we described a community of agents learning from each others, each having its own point of view and interpretation of events occurring in its environment, the learner refuting an EQ acting temporarily as the Teacher. Defining this interaction as a competition to optimize a score motivates the emission of queries.

This multi-agent discovery platform offers various industrial applications, especially as a tool for analysts trying to have a synthetic vision of a complex situation described by heterogeneous information sources, or for optimizing a production process involving complex systems.

References

1. Angluin, D., Krikis: Learning from different teachers. Machine Learning **51** (2003) 137–163
2. Nobrega, G.M.D., Cerri, Sallantin: A contradiction driven approach to theory information: Conceptual issues pragmatics in human learning, potentialities. Journal of the Brazilian Computer Society **9** (2003) 37–55
3. Angluin, D.: Queries and concept learning. Machine Learning **2** (1988) 319–342
4. Angluin, D.: Queries revisited. Theoretical Computer Science **313** (2004) 175–194
5. Popper, K.R.: Conjectures and Refutations: The Growth of Scientific Knowledge. Harper and Row (1963)
6. Cavaillès, J.: Sur la logique et la théorie de la science. Librairie Philosophique J. VRIN (1997)
7. Beziau, J.Y.: La logique paraconsistante. Logiques classiques et non classiques, essai sur les fondements de la logique (1997)
8. Nakamatsu, K., Kato, T., Suzuki, A.: Basic ideas of defeasible deontic traffic signal control based on a paraconsistent logic program evalpsn. Advances in Intelligent Systems and Robotics (2003)
9. Angluin, D.: Negative results for equivalence queries. Machine Learning **5** (1990) 121–150
10. L.Blum, Blum: Toward a mathematical theory of inductive inference. Inform. Control **28:2** (1975) 125–155
11. Gold, E.: language identification in the limit. Inform. Control **10** (1967) 447–474
12. Sallantin, J.: La découverte scientifique assistée par des agents rationnels. Revue des sciences et technologie de l'information (2003) 15–30
13. Liquière, M.: Structural machine learning with galois lattice and graphs. International Conference on Machine Learning - ICML (1998)
14. Chavalarias, D.: La thèse de Popper est-elle réfutable? Memoire de dea, CREA - CNRS/Ecole Polytechnique (1997)
15. Gardner, M.: Mathematical games. Scientific American (1959)
16. Cole, M., Engeström, Y.: A cultural historical approach to distributed cognition. Distributed Cognition (1993) 1–46
17. Garland, A., Alterman, R.: Multiagent learning through collective memory. Adaptation, Coevolution and Learning in Multiagent Systems: Papers from the 1996 AAAI Spring Symposium (1996) 33–38
18. Dunbar, K.: How scientists really reason: Scientific reasoning in real-world laboratories. Mechanisms of Insight (1995)