Learning of agents with limited resources – extended version

Sławomir Nowaczyk
Department of Computer Science
Lund University
Box 118, 221 00 Lund, Sweden
Slawomir.Nowaczyk@cs.lth.se
http://www.cs.lth.se/home/Slawomir_Nowaczyk/
4th February 2005

Abstract

In this paper we present our preliminary investigation of rational agents who can learn from their experience. We claim that such agents need to combine at least three attributes – deductive reasoning based on currently possessed knowledge, inductive learning from their history and awareness of passing time (including the necessity to commit computational resources to tasks which need to be carried out). We analyze how those aspects interact and initiate discussion of whether it is feasible, with current technology in each of the related fields, to combine them in a valuable way. More detailed description of this research can be found at http://www.cs.lth.se/home/Slawomir_Nowaczyk/phd-desc/ This document is a more elaborate version of extended abstract I have submitted to AAAI-05 SAAP and of thesis summary I have submitted to AAAI-05 DC.

Introduction

The basic idea of this project is to investigate a methodology for developing rational agents – both virtual and physical ones – that would be able to learn from experience and to improve in task they are designed to solve.

A rational agent should use deductive reasoning to take advantage of whatever domain knowledge it has been provided with. Besides that, it is expected to perform inductive learning in order to benefit from experience and correct missing or inaccurate parts of that knowledge. Finally, it must acknowledge the fact that both reasoning and acting takes time and try to balance those activities in a reasonable way.

In other words, agents we are discussing in this work are supposed to combine deductive and inductive reasoning with time-awareness. The interactions among those three aspects seem to be crucial for developing truly intelligent systems and are, therefore, an interesting and promising area for further research.

Our work at this point is rather preliminary. We have just defined our expectations, some features and assets we would like our agents to possess, and we have done some investigations of what their consequences might be. A review of available technologies which might allow formalizing and implementing those features has only been initiated.

It is an obvious truth that in practical applications some amount of knowledge is available, about such things as the environment, agent itself, possible actions and their consequences, etc. On the other hand, it is usually neither as comprehensive nor as detailed as one would like. It is therefore very important for an agent to be able to make use of this knowledge as much as possible, while at the same time agent has to remain aware of its limitations and look for ways to confirm, expand or correct that knowledge.

One way of doing this is by taking advantage of experience, which clearly offers immense possibilities for improvement – by using, for example, inductive learning. It can be used, among other things, to direct deductive reasoning and to refine current knowledge. If an agent's life is relatively long in comparison to the duration of a single task, the benefits of learning are even more clear.

Any real agent has many resources that are limited, such as time and memory, and to be considered rational it must take those limitations into account. It is not our goal to consider strict deadlines or precise time measurements, but rather to express that a rational agent needs general time-awareness and ability to reason about committing various resources to various tasks, as suggested in (Chong *et al.* 2002). In particular, it is not justified to assume that an agent knows all deductive consequences of its own beliefs.

It is clear from the above that we need to carefully choose what kind of logic do we intend our agent to use for its reasoning. We need something that can handle incomplete and inconsistent facts in a reasonable way – since we intend to use inductive learning, we will be dealing with knowledge which can turn out to be incorrect. And since we discard the idea of omniscience, our agent cannot be expected to immediately discover that some two pieces of knowledge contradict each other.

Moreover, in order to be able to intentionally direct its own learning process, an agent must be able to reason about its own knowledge and lack of thereof – thus, the logic we choose must strongly support epistemic concepts. Ideally, it should have provisions for modeling agent's ongoing reasoning process in some incremental way.

Our first attempts to define and formalize an agent as described above call for an example problem. A reasonably nice one is the well-known game of Wumpus. Among its many variations there are some which are very simple and there are some which are quite complex. This suggests that this game can continue to remain challenging even as we progress in our research and agent's reasoning capabilities increase.

At the same time, all those variants share many similarities – especially on the conceptual level – which makes it much easier to talk about and analyze effects of improvements and modifications to agent's reasoning machinery.

In its simplest form, the game takes place on a square board through which an agent is allowed to move at will. One square is inhabited by the Wumpus, but player doesn't know which one is it. Agent's goal is to kill the monster by shooting an arrow onto the square it occupies, while avoiding getting eaten by the monster. Luckily, Wumpus is a smelly creature, so the player always knows if the monster is in one of the squares adjacent to his current position – but unfortunately not on which one.

The game can be easily modified – and complicated – by allowing the Wumpus to move, by providing more than one Wumpus, by changing simple square board into a maze, by introducing additional, different monsters, by providing an opportunity to search for treasure, etc. Also, there is a possibility of changing actions outcomes and various encounters into probabilistic events. If so extended, this game exhibits surprisingly many features which make real problems difficult, and thus gives us hope that methodology developed and tested on this example will be genuinely useful.

Currently we have progressed to a point in our research where we have theoretical background for a simple agent who is able to operate within a small world of the simplest Wumpus game. It is lacking the learning component, and it still does not have any capabilities to plan – so it cannot come up with winning strategy. It can play the game, though. Right now we are looking for a suitable implementation in order to automate agent's reasoning.

Solution

The basic form of Wumpus game clearly has a winning strategy for the player (barring obvious exceptions where Wumpus is located too close to his starting position). Finding it in an automatic way, however, is far from trivial.

Despite that, it is our goal to create an agent who – preferably given nothing more than game rules and some way of gathering experience – is able to discover such strategy. Moreover, we are interested in analyzing situations where agent's *a priori* knowledge of the rules is neither complete nor faultless. We sincerely hope that this research will result in development of a *methodology* for creating and improving an agent so it can deal with more complex variants of the Wumpus game and degrade gracefully when not enough domain knowledge or too little time is available.

Our approach is related to several well-established areas of computer science. Quite a few of them could be used to solve the Wumpus problem. On the other hand, there are also significant differences between our idea and those presented earlier, which we discuss below.

Winning a Wumpus game can be seen as a typical planning problem, and there are many more or less standard algorithms (Russell & Norvig 2003) within that area which could solve it, at least in its simple variants. However, our aim is to be able to tackle a wider range of problems, beyond those that can be comfortably expressed in the realm of planning – including ones with imperfect domain knowledge.

On the other end of spectrum are methods originating from inductive learning paradigm (Mitchell 1997), such as reinforcement learning (Sutton & Barto 1998), which typically do not use any *a priori* knowledge. We believe that this kind of approach would work fairly well for the simple Wumpus games, but the cost of ignoring available information seems too high in more complex variants. In that sense, it is not clear how to scale those solutions up to more complex cases. We are strongly convinced that symbolic reasoning is a necessity for kinds of problems we want to tackle.

The solution most similar to what we aim at are probably automated theorem provers. Our approach can be described, in a bit simplistic manner, as an attempt to amend a theorem prover with ability to learn inductively and to take time limitations into account. However, our agent needs to be able to reason in some logic capable of expressing epistemic concepts, since it has is supposed to consider its own knowledge and act accordingly. This poses additional problems, since while epistemic logics are well-explored (Fagin *et al.* 1995; Reiter 2001), there are surprisingly few systems that are able to efficiently reason in an automatic way using those formalisms.

Up to now many successful approaches have been presented that combine, in various ways, inductive learning with deductive reasoning (Bergadano, Giordana, & Saitta 1991; Shavlik & Towell 1989; Brodie & DeJong 2001; Laird & Rosenbloom 1996), which suggests it is an interesting and promising area of research. As far as we know, however, those ideas have not been applied to rational agents in the sense we detailed in Introduction.

Another distinctive characteristic of our solution is the notion of time-awareness. In order to achieve it, we intend to employ Active Logic (Elgot-Drapkin *et al.* 1999) as agent's underlying reasoning apparatus. This logic was designed for non-omniscient agents and additionally has mechanisms for dealing with contradictory and uncertain knowledge. We believe it is a good reasoning technique for versatile agents, especially since it has been successfully applied to several different problems – including some in which planning plays a very prominent role (Purang *et al.* 1999).

At the same time, in our attempts to develop general methodology, we would like to avoid reverting to the oldest and still the most efficient way of solving computational problems: hand-written programs, tailored by a computer scientist to a particular instance and using complex, specialized algorithms. Therefore it is our primary goal to remain reasonably domain-independent and to develop solutions as versatile as possible.

To summarize, we are aiming at a solution lying somewhere between the efficiency of planners and the generality of theorem provers, with an added ability to learn from experience. In this context, treating the Wumpus problem as an instance of a more general class of problems, we hope to find interesting and useful solutions, and appraise the territory for further research.

The logical framework we intend to use, Active logic, was developed as a means of combining inference reasoning with reactivity, and thus it contains a special *evolving-during-inference* notion of time. Its singular predicate $Now(i)$ keeps track of current time and how it progresses in parallel with agent's reasoning. Active logic is not so much a strict formalism as rather a hybrid of formal system and inference engine.

Due to those facts, it has several valuable and functional characteristics which are particularly suitable for kinds of agents we are interested in. One is the ability to integrate external knowledge into the reasoning process at any time, important to us since we expect our agents to achieve new information through learning and through observing the world. Another is the time-sensitivity, when a formula proven – and thus considered true – at time point i is always *remembered* as being believed true at that point, but does not have to *remain* being viewed as such at later time. This allows for informed and sensible handling of inconsistencies, with agent being able to reason about their sources and ways to correct them.

To formalize the notion, Active Logic can be seen as a language of first order logic in which formulae are indexed by time at which agent considers them to hold. There are also special inference rules aware of this indexing, which direct the reasoning process. Most of them correspond to standard rules of inference, but there are some special ones. One of the most prominent is knowledge inheritance rule, which – in absence of contradictions or similarly unusual situations – makes sure that every formula considered true at time i is also considered true at time $i + 1$. Another is the time progress rule, which in every step from fact $Now(i)$ infers $Now(i + 1)$.

Our toy problem is simple enough to allow easy analysis of both agent's reasoning process and solutions found, also by hand. At the same time it seems to be scalable enough to provide experimental grounds for comparing different implementations and to allow checking whether our methodology is able to find non-trivial solutions.

Methodology

We assume that the agent is given, as an input, two things: some representation – possibly incomplete – of the Wumpus game rules, and some kind of environment, like a simulator, which can be used to gather experience. By utilizing them, the agent should find – and justify – a strategy for playing the game.

An important point to observe is how in this project we are going to analyze inductive learning which differs in two main aspects from what is usually thought of as typical in this domain. First, we are interested in giving our agent an opportunity to *intentionally* direct its learning process – it should, on the basis of its current knowledge, identify suitable time and method of learning, as well as the concept it is trying to learn about. It should choose between, for example, verifying a hypothesis it may already consider plausible and attempting to discover more general laws of the world. Second, we are looking for a way to use deductive knowledge, in addition to past experience, as a guiding force of the learning process.

In particular, when validity of some formula needs to be established, an agent is supposed to evaluate various methods of doing so – deduction can be one of them, but simply acting based on this formula and analyzing the outcome of such action may be, in many situations, preferred. We are interested in investigating whether this is indeed true and how can we make an agent recognize such situations.

A somewhat simplified view of agent's behavior could be described by the following steps, performed in a continuous loop:

- Reason deductively while it is providing satisfactory results and sufficiently enriches knowledge base.
- Determine what should be learned, design inductive learning experiments which are likely to achieve that goal and perform them.
- Analyze results of those experiments and assimilate new information, integrating inductive and deductive knowledge.
- Determine if a satisfactory solution has been found.

Basically all of the steps described above are very important and highly non-trivial. Since we are still in a preliminary phase of the research, we can only point out to some problems and interesting aspects in several of them.

One of the most important points is how to decide when deductive reasoning is not providing satisfactory results. It is necessary, but at the same time quite difficult, to find a better measure than number of inference steps or elapsed time.

Having an agent figure out by itself what should it attempt to learn about and how to perform experiments which would yield appropriate results is an extremely interesting topic in itself – especially if we consider interactions with user to be an option, and we see no reason to do otherwise. There has been some research in this area of learning, but not many results have been achieved, especially not in case of experience having form of structured data or logical formulae.

Finally, it is not obvious how an agent should know it has completed its task in a satisfactory manner. For our simplest Wumpus game, how is it to know that a winning strategy has been found? In some situations, when domain knowledge is rich enough, this fact *can* be deduced – but it may not be possible and feasible in every case.

As we mentioned earlier, several researchers have investigated the ways of integrating deductive and inductive knowledge and reasoning, and it is not clear whether one of their approaches can be extended to handle our case or if we will be forced to develop something different.

We expect our use of Active Logic to help us significantly in this respect. Several of its features, such as ability to smoothly integrate new knowledge – like the one coming from experience – at any time, its ability to handle inconsistencies and reason about passing time seem to be crucial for our needs.

Our next step is to implement a prototype system which would be able to perform the kind of reasoning we have discussed up to now. We also intend to find another problem, beyond the Wumpus game, which we could evaluate our solution against. One such problem, very promising, seems to be general game playing – where a system is given rules of an arbitrary game, described in a formal language, and is supposed to play this game without any human intervention.

Example

Let us consider one possible axiom of the Wumpus game. For simplicity of presentation we are ignoring at this point the time index of Active Logic and we omit the relation between agent's knowledge and its *current* situation and *future* plans it may be considering:

$$K[\, smell(a) \leftrightarrow \exists_x (Wumpus(x) \wedge Neighbour(a, x))\,]$$

Due to limited space in this summary we are unable to explain the more technical details – interested reader can find them on author's web page – but intuitively this axiom means: *agent knows it smells on exactly those squares which neighbour Wumpus' position*. It seems to be pretty straightforward way to formalize one of the rules of our game. Notice how we implicitly assume here a non-moving Wumpus, as we do not index $Wumpus$ predicate by time in any way.

From this, using reasonably natural inference rules, an agent can – within only a couple of inference steps – deduce following rule:

$$\forall_x \; K[\, \neg smell(a) \wedge Neighbour(a, x)\,] \leftrightarrow \\ K[\, \neg Wumpus(x)\,]$$

Clearly, an agent does not know whether it smells on the squares it has not yet visited. We are also assuming that it has a reasonable notion of geometry – in particular, it knows what *Neighbour* means.

Nevertheless, with this formula, an agent can – as soon as it does visit a non-smelling square – immediately reduce its uncertainty about location of the Wumpus. Moreover, it is able to reason about knowledge it will – or could – gain by performing various actions.

We have not worked out the details of inductive learning component yet, but we expect it to be able to discover formulae similar to the following one:

$$K[\, player(a) \wedge smell(a) \wedge unexplored(b)\,] \rightarrow \\ K[\, \neg shouldMove(b)\,]$$

Intuitively this formula means that it is dangerous to move onto an unvisited square if it smells at player's current position. Having an inductive learning system capable of discovering such a rule seems quite likely, as there are several known machine learning algorithms that are – in theory, at least – able to achieve this.

Of course, it is – in principle – possible to deduce the above formula from the rules of the game, but it is evidently going to be a complicated process. In addition, if agent's knowledge is incomplete (for example, if it doesn't know the exact conditions for loosing the game), deductive reasoning will fail to produce such formula, while it may still be learned.

The interesting part is that those two formulae should basically enable an agent, with a bit of additional technicalities, to determine Wumpus' position – which essentially allows an agent to win the game, or at least the simplest variant of it.

One of the technicalities we have omitted in the above descriptions is the fact that world's description *changes* in time, at the very least as a result of agent's actions. Thus, it is necessary for an agent to be aware, for example, that while it did not know at time $i - 10$ whether it smells on square x, it knows now that it does. And even if it does not know right now if it smells on square y, it *will* know this in next step *if* it performs some particular action.

Thus, it seems to be necessary to take into account both the past history of the game, and the course of action agent is currently considering, when formalizing its knowledge.

Conclusions

We are well aware that at this preliminary stage of research it is difficult to predict how useful our approach can be. Nevertheless, the intuitions on which we base our considerations seem sound, and the fruitful work of several scientists in related areas appears to bode well for this project. On the other hand, we are not aware of anybody who has attempted to combine all three aspects we are interested in: deduction, learning and time-awareness, the way intend to do it.

There is a lot of problems which remain unsolved and many unanswered questions await careful consideration. However, at this stage, learning of agents with limited resources seems to be an interesting area for further research.

Right now our main goal is to create a prototype system which would allow us to verify the feasibility of our ideas. If it works well, we may even be able to participate in the AAAI First Annual General Game Playing Competition

References

Bergadano, F.; Giordana, A.; and Saitta, L. 1991. *Machine learning: An integrated framework and its applications*. New York: Ellis Horwood.

Brodie, M., and DeJong, G. 2001. Iterated phantom induction: A knowledge-based approach to learning control. *Machine Learning* 45:45–76.

Chong, W.; O'Donovan-Anderson, M.; Okamoto, Y.; and Perlis, D. 2002. Seven days in the life of a robotic agent. In *GSFC/JPL Workshop on Radical Agent Concepts*.

Elgot-Drapkin, J.; Kraus, S.; Miller, M.; Nirkhe, M.; and Perlis, D. 1999. Active logics: A unified formal approach to episodic reasoning. Technical Report CS-TR-4072, University of Maryland.

Fagin, R.; Halpern, J. Y.; Vardi, M. Y.; and Moses, Y. 1995. *Reasoning about knowledge*. MIT Press.

Laird, J. E., and Rosenbloom, P. 1996. The evolution of the Soar cognitive architecture. In Steier, D. M., and Mitchell, T. M., eds., *Mind Matters: A Tribute to Allen Newell*. Mahwah, New Jersey: Lawrence Erlbaum Associates, Inc. 1–50.

Mitchell, T. M. 1997. *Machine Learning*. McGraw-Hill Higher Education.

Purang, K.; Purushothaman, D.; Traum, D.; Andersen, C.; and Perlis, D. 1999. Practical reasoning and plan execution with active logic. In Bell, J., ed., *Proceedings of the IJCAI-99 Workshop on Practical Reasoning and Rationality*, 30–38. Murray Hill, New Jersey: International Joint Conference on Artificial Intelligence.

Reiter, R. 2001. *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems*. The MIT Press.

Russell, S., and Norvig, P. 2003. *Artificial Intelligence: A Modern Approach*. Prentice Hall Series in AI, 2nd edition.

Shavlik, J. W., and Towell, G. G. 1989. Combining explanation-based and neural learning: An algorithm and empirical results. Technical Report CS-TR-1989-859, Computer Science Department, University of Wisconsin.

Sutton, R. S., and Barto, A. G. 1998. *Reinforcement Learning: An Introduction*. Cambridge, MA: The MIT Press. A Bradford Book.