

Active Learning in Correcting Domain Theories: Help or Hindrance?

Douglas J. Pearson
Artificial Intelligence Laboratory
The University of Michigan
1101 Beal Ave.
Ann Arbor, MI 48109
dpearson@umich.edu

Abstract

In active learning, the learner uses its current knowledge to guide the choice of future training instances. This dependency between the agent's knowledge and its subsequent training set provides the active learner with a possibility for improved learning and at the same time a potential pitfall that can hinder learning. The opportunity for improved learning comes from choosing closely related instances, in such a way that they help the learner identify why one instance is positive and another is negative. The potential problem is that incorrect early learning can inhibit the learner from seeing instances that will lead it to correctly learn the target concept. These two problems are explored within the task of learning domain knowledge. I propose a method for avoiding the problem and taking advantage of the opportunity, with experimental evidence to verify that this improves the performance of a particular active learner, IMPROV.

Introduction

An autonomous agent that learns from its experiences of executing actions in an environment is learning *actively*. This is in contrast to passive learning, where a set of training examples are selected in advance by a human instructor or domain expert. Agents that learn about the effect of executing different sequences of actions, or plans, are learning or revising their domain knowledge. Existing approaches to learning domain knowledge can be divided into two main classes.

First, theory revision systems (EITHER (Ourston & Mooney 1990), OCCAM (Pazzani 1988; 1991), CLIPSR (Murphy & Pazzani 1994)) that reason deliberately about the correctness of the agent's domain knowledge. These systems typically learn in a purely passive manner. A set of training instances are selected in advance and are unrelated to the state of the agent's knowledge during training. The second class of domain theory learning systems are reinforcement learners (Q-learning (Watkins & Dayan 1992), Classifiers (Holland 1986), Backpropagation (Rumelhart, Hinton, & Williams 1986)). These systems use a general purpose induction algorithm applied to a procedural domain theory. They do not reason explicitly about the

correctness of the domain knowledge, rather they focus on the task performance and reward. Reinforcement learners are often used actively. The agent selects a plan based on its current knowledge, leading to a certain reward, which then forms the next training instance for the agent. The training instance is therefore dependent on the agent's current knowledge. This dependency of training instances on the agent's knowledge is at the heart of the problem and opportunity that are available for an active learner, as will be discussed in the rest of this paper.

	Passive Learning	Active Learning
Deliberate Reasoning about K	Theory Revision Systems	IMPROV
No Deliberate Reasoning about K		Reinforcement Learners

Figure 1: The space of systems learning domain knowledge

The research reported here focuses on a hybrid system, IMPROV (Pearson & Laird 1995), (see Figure 1), that attempts to draw on the strengths of both passive theory revision systems and active reinforcement learners. An IMPROV agent learns actively from its experiences of executing plans in a domain. The agent's domain knowledge is represented procedurally (to ensure efficient learning and tractable access to the knowledge) but IMPROV reasons deliberately about the correctness of the agent's knowledge (to achieve faster, higher quality learning than reinforcement learners). In this paper, I discuss the problem of incorrect early learning and the benefit of careful selection of instances that are associated with any active learning system. I will use IMPROV and the task of learning domain knowledge as an example of one approach to overcoming the problem and also one approach to taking advantage of the opportunity.

Active Learning Overview

This section formally describes the distinction between active and passive learning (as I am using the terms). The intention is to identify the important dependencies between the agent's knowledge and its training set.

Instances are selected from the set $ISPACE$, each instance being a plan and a label indicating success or failure of that plan. The nature of $ISPACE$ would naturally be dependent on the domain.

For any learner the agent's knowledge during training is given by :

$$K = \{K_0, K_1, K_2, \dots\}$$

and the set of training instances for the agent are :

$$I = \{I_1, I_2, I_3, \dots\} \text{ where } I_j \in ISPACE$$

For a passive learner, the set I of instances is present before the agent starts to learn. The relationships between the passive learner's knowledge and the instance set are given by :

$$K_j = PL(K_{j-1}, I_j)$$

I fixed and independent of K

That is, the agent's knowledge after seeing an instance is a function (PL , the passive learning algorithm) of the agent's knowledge before seeing that instance (K_{j-1}) and the new instance (I_j).

For an active learner, the set of instances are selected by the agent and therefore the relationships between the active learner's knowledge and the instance set are given by :

$$K_j = AL(K_{j-1}, I_j)$$

$$I_j = P(K_{j-1}, S_{j-1})$$

That is, the agent's knowledge is a function of its previous knowledge and the new instance. Additionally each instance is a function of the agent's knowledge and the current world state (S_{j-1}). In learning domain knowledge, this function, P , might be a planning method.

The observation here is that in active learning, both the agent's knowledge and the instance set depend on the previous state of the agent's knowledge. This interdependency leads directly to the problem of incorrect previous learning and the opportunity of selecting related instances during training. I will discuss these issues in more detail in the following sections.

The problem of incorrect previous learning for active learners

The problem for an active learner occurs when the agent makes an incorrect generalization during its early learning. The danger is that the incorrect learning will move the agent to explore a part of the instance space which is of little use in learning the target concept.

This is probably best seen through an example. In trying to learn to drive a car across an intersection, a training instance consists of a plan for crossing the intersection and whether that plan succeeds or not. Let's assume the concept to be learned is that it is alright to drive through an intersection when the light

is green and your speed is less than 30mph. For the first instance, the agent drives through the intersection and then receives feedback that it failed to cross correctly (perhaps it's pulled over by the police). Let's assume the light was green and the car's speed was 40mph. If the agent has no causal knowledge about the domain, then even in this trivial example, it is a difficult credit assignment problem to identify the cause of the failure as being the speed of the car. There is simply not enough information available for the agent to correctly determine the cause of the failure.

In such a situation, a learner will typically make an inductive guess about the cause of the failure. Let's assume it guesses the cause was that the light was green. This incorrect guess will naturally lead to lower performance on the task for either a passive or active learning system. However, they differ in their ability to recover from the incorrect learning. In a passive learner, the next training instance is chosen at random. Therefore it is no more, and no less, likely to see a counter example (successfully crossing when the light is green) than it was before it's initial incorrect learning.

However, in an active learner, the agent now believes that driving through an intersection when the light is green will lead to a failure. As a result, it will be less likely to attempt any plan that involves crossing the intersection when the light is green. The initial incorrect learning will adversely affect the future training set. The next time the active learner is faced by a green light, it is likely to wait until the light turns red before trying to cross. The general property is that it is difficult for the active learner to realize that it's initial learning was incorrect as it will tend to avoid counter-examples. It's previous learning leads it to explore a less useful part of the instance space (in this example, avoiding instances where the light is green) and therefore hindering its future learning.

Approaches to overcoming the incorrect previous learning problem

One approach to overcoming the problem of incorrect previous learning is to incorporate an exploration function into the use of the learned knowledge. Typically, this takes the form of probabilistically deciding whether to take the plan suggested by what the agent has learned or selecting another plan at random. This is the approach taken by many reinforcement learning algorithms. The approach is summarized in figure 2, where new knowledge is learned aggressively, but that knowledge is only used cautiously.

I have explored a different approach in IMPROV. Rather than learning after seeing each new training instance, IMPROV delays training until a set of instances has been collected. The size of the set is defined by the number of instances needed to find a positive instance (I'm assuming positive instances are rarer than negative ones which is usually true in planning tasks). By only training once a set of instances has been collected,

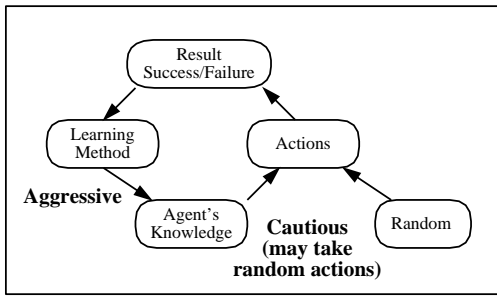


Figure 2: Cautious use of the agent's knowledge

the agent is more likely to identify the correct cause of the failure. To return to the driving example, having failed when crossing at 40mph, the IMPROV agent will avoid using that precise plan again, but it will not generalize the reason. It will therefore have no reason to avoid other plans which involve driving through a green light. Once it sees a positive instance (light is green, speed is 30mph) then by comparing this instance with the negative instances (light is green, speed is 40mph), the inductive learning can be biased towards the differences and so is more likely to identify the correct cause (i.e. high speed leads to a failure). As this learning is more likely to be successful, IMPROV places more confidence in the results and uses the learned knowledge more aggressively than a reinforcement learner would. This approach is summarized in figure 3, where new knowledge is learned cautiously but then that knowledge is used aggressively.

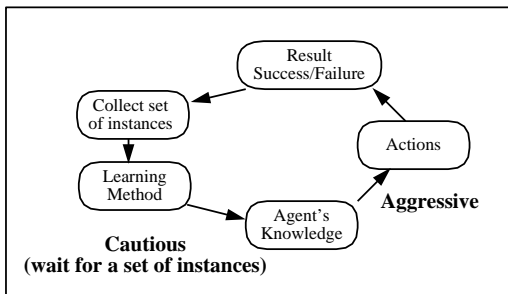


Figure 3: Cautious learning of the agent's knowledge

The opportunity to learn from selected training instances

The dependency of an active learner's future training set on its current knowledge produces benefits as well as problems for the learner. One such benefit, is that the agent can select instances which are closely related during training. The correct choice of training instances can make learning easier (VanLehn 1987).

To return to the driving example, let's assume the agent fails to cross an intersection when the light is red and it's driving at 30mph. If the agent is learning

actively, then the next time it comes to an intersection and the light is green it can choose the speed to use when crossing. For instance, crossing at 30mph is probably preferable over crossing at 20mph, if the agent wants to maximize the quality of its learning. When it discovers that crossing at 30mph with a red light leads to a failure, but crossing at 30mph with a green light leads to a success, it is clear that the red light caused the failure. A passive learner might be presented with an instance of crossing at 20mph with a green light, in which case the cause of the failure could still either be the speed (the speed limit is 25mph) or the color of the light.

IMPROV again takes advantage of this property by delaying its learning and only training once a set of instances has been collected. As IMPROV actively chooses which plans to execute, it chooses plans that are closely related to each other. Once an incorrect plan has been discovered, IMPROV incrementally explores the region of plan space surrounding the incorrect plan. This incremental exploration ensures that when a successful plan is found, it is closely related to the incorrect plans. This ensures that there are only a few differences between the correct and incorrect plans, which focuses the learner's attention on the important, causal, features.

There are many alternatives to IMPROV's scheme of searching related plans until it finds the first correct one. One alternative is to search until a unique cause of the failure is identified. In this case the active learner becomes an experimentation system, running experiments (i.e. closely related training instances) to determine the cause of a failure. Another alternative is to avoid plans which are similar to the failed plan. This would make learning more difficult, but would be appropriate if the agent is more concerned with quickly finding a solution to its current goals, than trying to maximize its learning. This might be seen as the difference between scientific research (where the learning is the main goal) and engineering (where the product is the main goal and learning is a side-effect).

Experimental Results

To verify that IMPROV's method, of collecting instances into sets before training, produces improvement on an active learning task, I ran comparisons between IMPROV and a modified version of IMPROV where the agent learns immediately after seeing each training instance. This modified version is more prone to the problem of incorrect early learning and does not take as much advantage of exploring the space of closely related plans. The experiment was run on a more complex version of the driving task described earlier and is described in more detail in (Pearson & Laird 1995).

The graph in Figure 4 show the cumulative number of errors made by each system during a series of different performance trials. Each trial consisted of

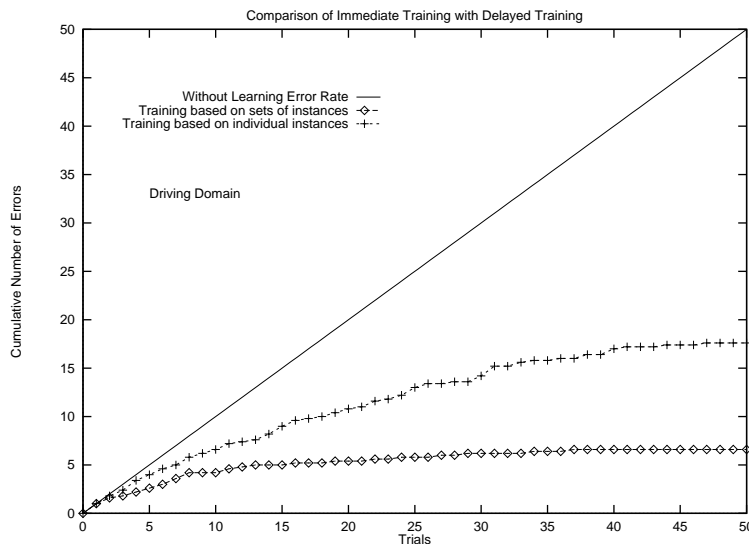


Figure 4: Comparison of immediate training to training on a set of instances

the agent attempting to cross an intersection. The agent was given an initial domain theory which was constructed to ensure that it would always make an error crossing any intersection. The diagonal line indicates this base-line error rate, which is the behavior of the system without any learning. The agent has partial control over its training instances as the nature of the intersection is fixed, but the agent is free to drive the car through the intersection according to any plan it chooses. The graph shows that delaying training produces a substantial improvement in the agent's performance as fewer errors are made. This shows the benefit of waiting until a set of related training instances has been collected and only then changing the agent's knowledge.

Conclusion

Active learning presents an agent with the opportunity to improve the quality of its learning through careful selection of its training set. However, early incorrect learning can lead the agent away from the parts of the search space that are necessary for correct learning. The proposal in this paper is that by being more cautious, and delaying training, an active learner can focus its learning more effectively and avoid making mistakes which will be harmful to its future learning. In this type of active learning system, it is more important to learn correctly than it is to learn quickly.

References

Holland, J. H. 1986. Escaping brittleness: The possibilities of general-purpose learning algorithms applied to parallel rule-based systems. In Michalski, R. S.; Carbonell, J. G.; and Mitchell, T. M., eds., *Machine*

Learning: An artificial intelligence approach, Volume II. Morgan Kaufmann.

Murphy, P. M., and Pazzani, M. J. 1994. Revision of production system rule-bases. In *Proceedings of the International Conference on Machine Learning*, 199–207.

Ourston, D., and Mooney, R. J. 1990. Changing the rules: A comprehensive approach to theory refinement. In *Proceedings of the National Conference on Artificial Intelligence*, 815–820.

Pazzani, M. J. 1988. Integrated learning with incorrect and incomplete theories. In *Proceedings of the International Machine Learning Conference*, 291–297.

Pazzani, M. 1991. Learning to predict and explain: An integration of similarity-based, theory driven, and explanation-based learning. *Journal of the Learning Sciences* 1(2):153–199.

Pearson, D. J., and Laird, J. E. 1995. Toward incremental knowledge correction for agents in complex environments. In Muggleton, S.; Michie, D.; and Furukawa, K., eds., *Machine Intelligence*, volume 15. Oxford University Press.

Rumelhart, D. E.; Hinton, G. E.; and Williams, R. J. 1986. Learning internal representations by error propagation. In *Parallel Distributed Processing*, volume 1. Cambridge, MA: MIT Press.

VanLehn, K. 1987. Learning one subprocedure per lesson. *Artificial Intelligence* 31(1):1–40.

Watkins, C. J. C. H., and Dayan, P. 1992. Technical note: Q-learning. *Machine Learning* 8:279–292.