Exploration in Machine Learning

Michael P. Frank
Program in Symbolic Systems
Stanford University
Stanford, California 94305

December 2, 1990

Abstract

Most researchers in machine learning have built their learning systems under the assumption that some *external* entity would do all the work of furnishing the learning experiences. Recently, however, investigators in several subfields of machine learning have designed systems that play an active role in choosing the situations from which they will learn. Such activity is generally called *exploration*. This paper describes a few of these *exploratory learning* projects, as reported in the literature, and attempts to extract a general account of the issues involved in exploration.

1 Introduction

What is exploration? The term carries connotations of experimentation, of trying new behaviors to learn their outcomes, of striving to move into unusual but interesting situations. The crucial point is that exploring is being actively engaged in the information-gathering process, not just internally analyzing a stream of input.

In looking over machine learning research, I have seen that a large portion of the current research concerns itself with the design of learning systems that play a passive role in the learning process; this includes most classification learning.[1] Other paradigms allow the machine to take an active role in determining what information it will have available to learn from. An example is the "learning by discovery" research, in which the learning systems often conduct experiments to test their hypotheses. The discovery-learning viewpoint, however, often focuses on abstract realms of applicability such as mathematics or science (e.g. the AM [9] and BACON [8] programs). There is less work on what an active role in learning

[1] See Shavlik and Dietterich [16] for a good overview of classification learning.

would be like for general agents, operating in an everyday domain, instead of a mathematical or scientific one.

However, interest in exploration seems to be growing; in recent years papers on the subject have appeared in many diverse subfields of AI and machine learning. In this paper, I will discuss four representative perspectives on exploring from the literature. One is from a classical problem-solving perspective, one addresses concept learning, the third emphasizes robot motor learning, and the fourth focuses on map learning. There is an amazing diversity in the four approaches; on the surface, no two have very much in common, except for the presence of some kind of principled exploration. Section 6 of this paper will attempt to generalize the four approaches to present a broader view of exploration in terms of which the particular approaches can be described.

2 Problem Solving

In the paper, "Rule Creation and Rule Learning through Environmental Exploration," [18], Shen and Simon describe a system for "learning from environment," which integrates exploration, learning, and problem-solving in a single framework. As would be expected from Simon, the work is totally within the classical AI tradition of logic-oriented problem-solving and planning; their program is based on STRIPS and GPS. Like any perspective, traditional AI emphasizes certain concerns and ignores others. For instance, it ignores uncertainty; later we shall say more about this. For now, let's look at the approach described in the paper.

Shen and Simon define the generic task domain as one in which there is a well-defined set of actions, percepts, and functional/logical contructors for building sentences in logic. There must also be an environment that exececutes actions and provides resulting percepts, and there must be a given goal state of the environment. These are all parts of standard task definitions for agents doing problem-solving, such as in Genesereth's work [5]. The difference is in their algorithm; their procedure alternates between generating an "exploration plan," learning from the results of the exploration, and attempting straight problem-solving. The learning phase involves a fairly straightforward generation and specialization process, and can be thought of in terms of Mitchell's version spaces [10] and related work. The problem-solving phase is simple STRIPS-style planning. The novelty lies in the creation of the exploration plans, which are intended to provide "suprising" experiences to serve as the basis for new rules. This process is reminiscent of the creation of "practice problems" in Mitchell's heuristic problem solver, LEX [11], except that Shen and Simon's work is focused on the creation of new pieces of the domain theory and not just heuristics for guiding planning.

The design of Shen and Simon's algorithm of course depends on their definition of the exploration problem. They give their system a single, well-defined goal, which is constant and known to the system during its entire exploration. Their exploration algorithm makes sense only given this state of affairs. Such background assumptions about the nature of goals turn out to be a central theme in exploring, which I will discuss later.

3 Classification Learning

In constast, the next paper, "Learning Novel Domains Through Curiosity and Conjecture," [15] by Scott and Markovitch, assumes that all the exploration takes place before any specific goals are known. This training-testing distinction reveals the classification/concept-learning perspective that Scott and Markovitch take. They define the exploration task for their program, DIDO, as follows: the domain is made up of "entities," which have various attributes; the agent can perceive the attributes, and perform "operations", which probabilistically change the values of the attributes. The problem-solving tasks in this domain presumably involve producing some goal entity, given an initial set of entities. However, the idea of there being multiple entities in the environment is totally irrelevant to the paper. The domain of the mapping to be learned is just the set of possible $(entity, operation)$ pairs, and the range of the mapping is a probability distribution on entity attributes. The actions that DIDO may perform during learning are simply the selection of any possible $(entity, operation)$ pair to be the next training instance. All environment situations are immediately available, whereas Shen and Simon's system had to plan to get into new situations. Thus, DIDO isn't realistically applicable to more reactive agent environments and embedded systems. [2]

However, what is interesting about DIDO is its particular technique for choosing training instances. It always chooses examples for which its internal representation is most *uncertain* about the outcome. So in DIDO, uncertainty is the driving force behind exploration. Because there is a probability distribution associated with the outcome of the examples, Scott and Markovitch are able to use Shannon's uncertainty function to calculate the uncertainty of the outcome precisely. The theoretical soundness of this measure indicates one advantage of using probabilty to represent learned knowledge when uncertainty is involved.[3] The vague goal of "exploration" can be quantified nicely in terms of reducing uncertainty about the environment, and Scott and Markovitch do so.

4 Robotic Manipulators

Andrew Moore's recent paper, "Acquisition of Dynamic Control Knowledge for a Robotic Manipulator," [13] resembles the previous paper in two ways: it treats the environment as providing a fixed-size vector of attributes, instead of a variable-size database of propositions, and it uses probabilities in choosing candidate actions for exploring. However, Moore's work resembles Shen and Simon's in that it presumes the existence of a precisely-defined goal to refer to during exploration, such as keeping a robotic manipulator positioned over a moving point by applying the proper torques to the arm's joints. Instead of choosing the action whose outcome is most uncertain, Moore's system chooses the action whose outcome, whatever it is, will most likely achieve the goal.

[2]See Leslie Kaelbling's Ph.D. thesis [7] for information on "Learning in Embedded Systems".

[3]Peter Cheeseman [1] has made a good case for the use of probability in machine learning.

At first it may be unclear how this method accomplishes exploration; won't it just choose whatever previous action failed the least, and not generate new actions? The answer is no, because the correct probabilistic formulas automatically give a better chance of success to actions that have not been tried extensively, as opposed to ones that have been tried and have failed. Thus, Moore's system searches new areas of the environment, until the correct action is found. [4] Barney Pell uses a similar "chance of success" criterion for his explorative learning program for the game of go [14]. Moreover, in both of these programs, the exploration is not really kept separate from the learning; the single strategy of choosing "most likely to succeed" actions seems to accomplish both the selecting of unexplored states and the achieving of the current goal based on the past history. However, this kind of exploring depends on there being a single, known goal; more will be said about this later.

5 Map Learning

Dean and Basye's system, described in "Coping with Uncertainty in a Control System for Navigation and Exploration,"[4] is like DIDO in that it does not depend on the presence of a single, specific goal. Instead, Dean and Basye characterize the problem as having both "immediate and anticipated tasks." The paper notes that, in general, acting to learn information for the benefit of future performance (i.e., exploring) is in conflict with acting to solve the current task.[5] Dean and Basye put forth that this is a problem for decision theory to resolve, and they describe several decision-theortic models for reasoning about the expected value of exploration. Again, probability plays a crucial role.

However, application of the decision theory turns out to be difficult, because a perfect solution would be computationally intractable. Approximations are used, and the map-learning problem is simplified to the problem of learning the connectivity of a rectangular grid of corridors. Even then, the analysis is not simple. This work seems like a case study in how probabilistic methods seem correct, but can often be hard to apply. However, probabilistic methods have had some interesting successes in machine learning, such as in Moore's system and in Cheeseman's work on Autoclass [3, 2]. Overall, the development of probabilistic methods seems important, but there is still a lot of work to be done in that area.

6 Exploring

What can we say about exploring, now, after having seen four such different descriptions of the exploration problem and its solution? Well, one common theme was the relationship of the goals to the learning situation. The various approaches differ in whether the goal is

[4] This presentation of Moore's system is simplified; in actuality, he has a continuous space of actions, and results from past actions are generalized smoothly to "nearby" actions in the space. Still, the basic ideas are the same.

[5] Pell notes this too, in [14].

known at the time of exploration, and whether more goals are expected to arise after the current one is achieved.

Any fully defined task for machine learning, i.e., any proposed domain, will have an associated space of possible goals, together with a temporal distribution for the goals in it. In Shen and Simon, this space consists of a single ever-present goal. In Scott and Markovitch, the goal-space consists of all possible entity-classification goals, but with the goals being indefinitely far away in the future. In Moore, there is a different spatial-location goal at each moment. And in Dean and Basye's simple decision-theory model, there may be a current spatial-location goal, together with other goals indefinitely far in the future.

Clearly, a good explorative learning program will have to take into account the goal-space of its environment. The four programs described above will only work in environments having the appropriate goal-spaces. These programs all have a kind of built-in bias, a bias towards dealing with a particular goal-space. We can call this the *goal-space bias*, and think of it as a specialized kind of bias similar to the inductive bias introduced by Mitchell [12].

By analogy with Mitchell's work, I belive it is possible to show that *exploration is meaningless or impossible without a goal-space bias*, although to prove this, exploration and goal-space bias would need to be defined formally, which is beyond the scope of this paper.

7 Conclusion

It seems that the problem of creating systems that explore, and explore well, can be seen as the attempt to implement a desired goal-space bias. The ideal exploration system, ignoring feasibility, would determine at each moment the action that would maximize expected performance over the enire expected goal-space, taking into account both the action's effect on expected performance for the current goals, and also its effect on the agent's knowledge about the environment so as to produce better performance on the expected future goals. Each of the four existing exploration systems discussed above crudely approximates this ideal in its own way, for some particular goal-space bias, although the papers don't describe explicitly all the assumptions about the goal space. Talking explicitly about goal-space biases is recommended for future work on exploring, so as to clarify the reason for particular exploring techniques, and perhaps lead to a new understanding of exploration in general and its place in machine learning.

8 Acknowledgements

Thanks to Barney Pell, Kurt Konolige, Peter Cheeseman, Michael Genesereth, and Mark Torrance for helpful discussions.

References

[1] Peter Cheeseman. In defense of probability. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, pages 1002–1009, Los Angeles, CA, 18-23 August 1985. IJCAI.

[2] Peter Cheeseman. On finding the most probable model. In Jeff Shrager and Pat Langley, editors, *Computational Models of Scientific Discovery and Theory Formation*, chapter 3, pages 73–95. Unknown Publishers, 1989.

[3] Peter Cheeseman, James Kelley, Matthew Self, John Stutz, Will Taylor, and Don Freeman. Autoclass: A bayesian classification system. In Shavlik and Dietterich [17], pages 296–306.

[4] Thomas Dean, Kenneth Bayse, Robert Chekaluk, Seungseok Hyun, Moises Lejter, and Margaret Randazza. Coping with uncertainty in a control system for navigation and exploration. In *Proceedings Eighth National Conference on Artificial Intelligence*, pages 1010–1015, Boston, MA, 29 July - 3 August 1990. American Association for Artificial Intelligence, AAAI Press / The MIT Press.

[5] Michael R. Genesereth. A comparative analysis of some simple architectures for autonomous agents. Other publication data unknown, 1988.

[6] IJCAI. *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, Detroit, MI, 1989.

[7] Leslie Pack Kaelbling. *Learning in Embedded Systems*. PhD thesis, Stanford University, Stanford, CA, June 1990.

[8] Pat Langley, Herbert A. Simon, and Gary L. Bradshaw. Heuristics for empirical discovery. In Shavlik and Dietterich [17], pages 356–372.

[9] Douglas B. Lenat. The ubiquity of discovery. In Shavlik and Dietterich [17], pages 341–354.

[10] Tom M. Mitchaell. Generalization as search. In Shavlik and Dietterich [17], pages 96–107.

[11] Tom M. Mitchaell and Paul E. Utgoff. Learning by experimentation: Acquiring and refining problem-solving heuristics. In Shavlik and Dietterich [17], pages 510–522.

[12] Tom M. Mitchell. The need for biases in learning generalizations. In Shavlik and Dietterich [17], pages 184–191.

[13] Andrew W. Moore. Acquisition of dynamic control knowledge for a robotic manipulator. In Bruce W. Porter and Ray J. Mooney, editors, *Proceedings of the Seventh International Workshop on Machine Learning*, pages 244–252, University of Texas, Austin, TX, June 21-23 1990. Morgan Kaufmann.

[14] Barney Pell. Exploratory learning in the game of go: Initial results. In *Proceedings of the London Computer Game Playing Conference*, London, 1990. Unknown organization.

[15] Paul D. Scott and Shaul Markovitch. Learning novel domains through curiosity and conjecture. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence* [6], pages 669–674.

[16] Jude W. Shavlik and Thomas G. Dietterich. Inductive learning from preclassified training examples. In *Readings in Machine Learning* [17], chapter 2.1, pages 45–56.

[17] Jude W. Shavlik and Thomas G. Dietterich, editors. *Readings in Machine Learning*. Morgan Kaufmann, 1990.

[18] Wei-Min Shen and Herbert A. Simon. Rule creation and rule learning through environmental exploration. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence* [6], pages 675–680.