to appear in: *Handbook of Brain Science and Neural Networks*
Michael Arbib (ed.)

Exploration in Active Learning

Sebastian Thrun
Universität Bonn
Institut für Informatik III
Römerstr. 164, D-53117 Bonn, Germany
Phone: +49-228-550-373, FAX: +49-228-550-382
E-mail: thrun@carbon.cs.bonn.edu

1 INTRODUCTION

Research on machine learning has, over the last decades, produced a variety of techniques to automatically improve the performance of computer programs through experience. Approaches to machine learning can roughly be divided into two categories, passive and active, each making characteristic assumptions about the learner and its environment.

1.1 Passive Learning

In the passive learning paradigm, a learner learns purely through observing its environment. The environment is assumed to generate a stream of training data according to some unknown probability distribution. Passive learning techniques differ in the type of results they seek to produce, as well as in the way they generalize from observations. Common learning tasks are the clustering, classification, or prediction of future data.

Passive learning techniques can be subdivided into order-free and order-sensitive approaches. Order-free approaches rest on the assumption that the temporal order in which the training data arrives does not matter for the task to be learned. It is assumed that the training examples are generated independently according to a stationary probability distribution. The majority of machine learning approaches falls into this category. For example, unsupervised learning usually aims to characterize the underlying probability distribution or to cluster the data. Supervised learning, on the other hand, is concerned with approximating an unknown target function (conditional probability) from a set of observed input-output examples.

Passive learning has also been studied in order-sensitive learning scenarios, which are settings in which the temporal order of the training data carries information relevant to the learning task. This is the case, for example, if consecutive training examples are conditionally dependent on each other, and learning about these dependencies is crucial for the success of the learner. Time series prediction or speech recognition are examples for order-sensitive learning domains.

1.2 Active Learning

The active learning paradigm differs from the passive learning paradigm, in which the learner is a pure observer, by the learner's ability to interact with its environment. More specifically, the learner can execute actions, which have an impact on the generation of training data. The freedom to execute actions imposes an important challenge that is specific to active learning: *Which actions shall a learner generate during learning? How can a learner efficiently explore its environment?*

In active learning one can also distinguish between order-free and order-sensitive cases. Order-free active learning rests on the assumption that what is observed in the environment depends only upon the most recently executed action. Perhaps the best-studied approach of this kind is learning by queries (Angluin, 1988), (Atlas *et al.*, 1990), (Baum and Lang, 1991). In query learning, the available actions are queries for values of an unknown target function. The environment provides immediate responses (answers) to these queries.

In order-sensitive approaches, on the other hand, observations may depend on many actions. For example, approaches to learning optimal control (like airplane control, or game playing) fall into this category. To describe the long-term dependencies between actions and observations, it has frequently proven helpful to assume that the environment possesses internal state information. Actions influence the state of the environment, and the state determines what the learner observes.

Exploration refers to the process of selecting actions in active learning. Although most of the exploration techniques reviewed in this paper are applicable to active learning in general, we will primarily focus on action selection issues in order-sensitive scenarios. Indeed, most of the approaches listed here have originally been applied in order-sensitive frameworks. Notice that throughout the paper we will make the simplifying and restrictive assumption that the state of the environment is fully observable.

2 ACTION SELECTION STRATEGIES

2.1 Principles

How can a learner pick the right action for learning? At first glance, it might seem appropriate to use random action selection mechanisms to generate actions (Whitehead, 1991). Random action selection is frequently used, primarily for two reasons: (a) it is simple, and (b) it usually ensures that any possible finite sequence of actions will be executed eventually. However, it has been shown, both through theoretical analyses as well as empirical findings, that more sophisticated query and exploration strategies can often drastically reduce the number of training examples required for successful learning. This is because responses to different actions typically carry different amounts of information. Random sampling often does not take full advantage of the opportunity to select the most informative query/action.

Intuitively speaking, in order to learn efficiently one would like to execute actions that are most informative for the learning task at hand. The more one expects to learn from the outcome of an action, the better it is. Indeed, this "greedy" principle, the optimization of knowledge gain, has been employed in most approaches to action selection in active learning.

2.2 Query Selection

Recent research on query learning has led to a variety of approaches for the active selection of queries. For example, in (Atlas *et al.*, 1990), (Cohn, 1994) two approaches to learning by queries are described that both use a neural network model of the learner's uncertainty. During learning, queries are favored that have the least predictable outcome. Uncertainty is estimated either by the difference of two models constructed from the same observations (Atlas *et al.*, 1990), or based on an analysis of the parameters of the estimator (Cohn, 1994). Both approaches have proven superior to random sampling in empirical comparisons. (Paass and Kindermann, 1995) propose a method that integrates an external cost function into the active learning framework. More specifically, their approach favors queries that minimize the decision costs, which allows to focus learning on performance-relevant areas. Their approach is computationally expensive, since it relies on explicit Monte-Carlo integration.

2.3 Exploration

In order-sensitive scenarios, exploring unknown parts of the environment requires sequences of actions to be executed. For example, if a lunar robot agent aims to explore the back side of the moon, it has to get there first—and getting there might require even more exploration. Techniques that employ models of the expected knowledge gain and then direct explorative actions to unknown parts of the environment are called directed exploration techniques (see (Thrun, 1992b) for an overview). While most existing approaches share the philosophy of selecting actions through maximizing the knowledge gain, they differ in the particular way knowledge gain is estimated. Some estimate this quantity implicitly from a specific data structure, other utilize explicit models represented using separate data structures. In addition, a variety of heuristic estimators have been used for estimating the expected knowledge gain. Estimators typically use related quantities such as frequency, density, recency, or empirical prediction errors.

For example, (Kaelbling, 1993) suggests an approach to exploration in which actions are favored unless they have repeatedly been found to be disadvantageous. As a consequence, actions are selected that exhibit good performance or that are unexplored or both. A similar approach following the same line of thought is proposed in (Koenig and Simmons, 1993). Their approach bears close resemblance with heuristic search techniques for graphs, (Korf, 1988), although it differs in that it does not assume the availability of a model of the environment. Koenig and Simmons also derive worst-case bounds for the complexity of exploration for deterministic shortest-path problems. In (Sutton, 1990), a so-called exploration bonus is assigned to actions. This bonus measures, for each environment state, the elapsed time since each available action was executed. As a consequence, actions that were not executed for a long time are favored for exploration. Sutton also employs a dynamic programming technique to propagate exploration utility through the state space of the environment using a model of the environment, which is easy to obtain for the environments he studied. Another approach to exploration has been proposed by Dayan and Sejnowski (unpublished). Here exploration achieved through a Bayesian prior that expresses uncertainty as a function of how often and when an action has been executed. In (Thrun, 1992a) several of these approaches are compared empirically, along with a combined approach taking frequency and recency into account. Approaches specific to memory-based learning can be found in (Moore, 1990) and (Schaal and Atkeson, 1994). Memory-based learning memorizes all training data explicitly. In these approaches, the density of previous data points is used to asses the utility of actions for exploration. Schaal also takes into account knowledge about the goal of learning to focus exploration.

2.4 Focussing Exploration

Most active learning techniques estimate the expected knowledge gain of the learner for each applicable action, and they select actions through maximizing knowledge gain. In order for this methodology to work efficiently, two assumptions have to be made: (a) the heuristic for estimating the gain of knowledge must yield approximately correct action preferences, and (b) gaining knowledge per se must be helpful for the learning task.

Both assumptions are not necessarily fulfilled in practice. Most heuristics for exploration are somewhat ad hoc, hence their effectiveness varies across environments and learning tasks. Moreover, depending on what goal the learner aims to achieve, sometimes only parts of the environment have to be known in order to perform optimally. This is typically the case, for example, in the context of reinforcement learning (Barto *et al.*, to appear), (Sutton, 1990), (Watkins and Dayan, 1992). In reinforcement learning, the learning task is to generate control, i.e., to learn action policies that maximize a given reward function. Exploring regions in state space that are irrelevant for the task of learning control is a waste of both time and memory resources.

A common strategy to focus exploration is to explore and to exploit simultaneously, by taking both knowledge gain and the task-specific utility of actions into account. Boltzmann distributions and semi-uniform distributions provide ways to combine random exploration with exploitation. These distributions explore by flipping coins, but the likelihood of individual actions is determined by the task-specific exploitation utility: In Boltzmann distributions the likelihood of picking an action is exponentially weighted by its utility, and in semi-uniform distributions the action with the largest utility has a distinct high probability of being executed. Notice that most of the aforementioned approaches have indeed originally been proposed in combination with task-specific exploitation. In (Thrun, 1992b) it has been empirically demonstrated that the combination of exploration and exploitation can yield faster learning than either component in isolation. The fundamental dilemma of choosing the right ratio between exploration and task-specific exploitation is called the exploration-exploitation dilemma. Often exploration and exploitation are traded off dynamically so that exploration fades in time.

2.5 Complexity Results

In addition to empirical studies, theoretical results emphasize the importance of exploration in active learning. Based on a result in (Whitehead, 1991), which shows that random walk exploration can require exponential learning time in various cases, it has been shown that directed exploration techniques can reduce the complexity of active learning from exponential training time (random exploration) to polynomial training time (Thrun, 1992a), (Koenig and Simmons, 1993). Similar results exist in the query learning framework (Angluin, 1988), (Baum and Lang, 1991). Although most results apply to certain deterministic environments only, in practice they often carry over to stochastic environments.

3 EXAMPLE

This section briefly describes an artificial neural network approach to exploration in real-valued domains. This approach shares many of the ideas in the current literature: the expected knowledge gain is estimated during learning, and actions are selected greedily such as to maximize knowledge gain. Unlike most other approaches, it operates in real-valued domains and uses artificial neural networks to estimate the gain of knowledge.

3.1 Modeling Competence

Assume the learning task is to approximate an unknown target function $f : I \rightarrow O$. Here, I denotes the input space, and O denotes the output space of both f and its approximation, denoted by $\hat{f}$. Assume that at any instance of time, the learner can execute one of its actions, denoted by A. Actions influence the state of the environment and hence have some impact on the training examples given to the function approximator. Notice that in the experiments reported below I is the product of A and the state space of the environment.

A competence map is a function $\hat{g} : I \rightarrow \Re$ that assesses the accuracy of $\hat{f}$. It is trained as follows. For each observed input-output example $\langle i, f(i) \rangle \in I \times O$ of f, there will be some model error $\varepsilon(i) = ||\hat{f}(i) - f(i)||$. The competence map models this error $\varepsilon(i)$ as a function of i. Hence, each training example for $\hat{f}$ also produces a training example for $\hat{g}$, as illustrated in Fig. 1. The competence map is used to direct exploration, by selecting actions that maximize $\hat{g}$. More specifically, the learner explores by picking actions for which its own competence is minimal, i.e., for which its internal models are most inaccurate. Such actions are assumed to maximize the gain of knowledge.

It should be noted that competence estimates, as they are described here, may only be approximately correct, since the dynamics of the estimators $\hat{f}$ and $\hat{g}$ are usually hard to model. In addition, if due to model limitations $\hat{f}$ fails to model the environment in sufficient detail, unmodeled effects can be a constant source of model error and perpetually provoke more exploration. This is the case, for example, if (many-to-one) function approximators like artificial neural networks are employed in highly stochastic environments,

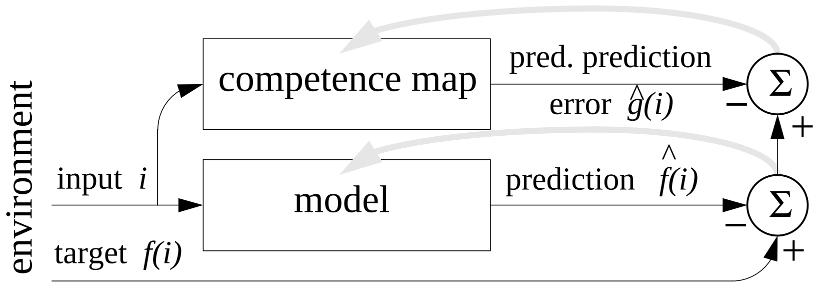

Figure 1: Training the model and the competence map.

3.2 Empirical Results

To illustrate exploration via a competence map, consider the environment depicted in Fig. 2. The input to the learner is its x-y-position in a two-dimensional world. Its task is to navigate from the starting position (box) to the goal position (cross) while avoiding collisions with the walls or the obstacle. Actions, denoted by $(\Delta x, \Delta y)$, are small displacements which, when executed, are added to the current position. When the agent reaches its goal or, alternatively, when it collides with a wall or the obstacle, it is reset to its starting position.

In addition to its coordinates, the learner is able to perceive a potential function $d(x, y)$, which is depicted in Fig. 2a. The potential measures the "distance" to the goal and to the obstacles, such that steepest descent yields a collision-free path to the goal location from arbitrary starting positions. Both the state transition function and the potential function are initially unknown. The goal of learning is to learn a control strategy for selection actions which carry the agent to the goal. This is done by learning the state transition function and the potential function. Once a reasonable model of these functions has been identified, pure hill climbing will result in admissible paths, such that the goal can be reached without collision.

In our experiments, a multi-layer network was trained with the Backpropagation algorithm to model the motion dynamics and the potential function values. The input to this network was the current position (x, y) and action $(\Delta x, \Delta y)$. It was trained to predict the next position (x', y') and the corresponding potential function value $d(x', y')$. The actual network consisted of two separate components, one for predicting the next position (x', y') with no hidden units, and one for predicting the potential function value. The latter component consisted of 10 units with radial-bases activation functions in the first hidden layer, and 8 units with a logistitic activation function in the second layer.

Competence was also modeled by an artificial neural network which received the same four input values as the model, but was trained to predict the squared model prediction error, given by $\alpha\ [(x'_{\rm pred} - x'_{\rm obs})^2 + (y'_{\rm pred} - y'_{\rm obs})^2 + (d_{\rm pred} - d_{\rm obs})^2]$. Here α is an appropriate normalization constant which ensures that competence values lie in $[0, 1]$. In the actual implementation,

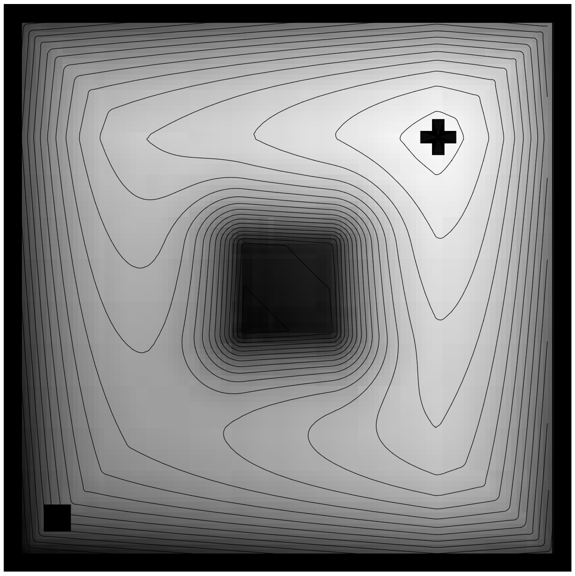

Figure 2: Potential function. The darkness indicates the combined “distance” to the goal and the obstacle/walls.

the competence network had two hidden layers with six logistic units each. During learning, exploration and exploitation were combined using a selective attention mechanism described elsewhere (Thrun, 1992b), which traded off exploration and exploitation dynamically based on expected costs and benefits.

In an experimental study three approaches to exploration were compared: (a) random exploration, (b) pure exploitation, i.e., always following the best known path, and (c) directed exploration based on competence. Since pure exploitation might get stuck and fail to explore exhaustively, in rare cases actions had to be generated randomly. In 15,000 learning steps, each technique learned the linear motion dynamics well. However, they produced different models of the potential function. Random exploration (c.f. Fig. 3a) performed most poorly. The resulting model was not accurate enough to allow the agent to navigate to the goal. When actions were generated by pure exploitation, a reasonable path was found from the start to the goal. The approach yielded good performance in terms of navigation. The model, however, was rather inaccurate and the world was poorly explored, as can easily be seen from Fig. 3b. The best results in terms of both control and model accuracy were found with directed exploration using the competence map. Competence map exploration also resulted in a much smaller number of collisions during learning, yet yielding the most accurate model. These findings demonstrate the advantage of directed exploration techniques over random exploration. Further details may be found in (Thrun, 1992b).

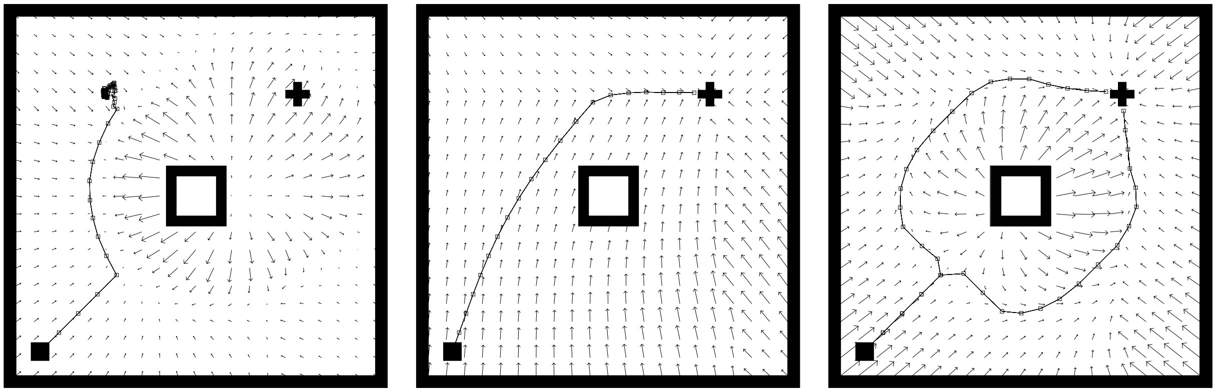

Figure 3: Results. (a) Random Exploration. (b) Exploitation (whenever possible). (c) Exploration with a competence map.

4 CONCLUSION

In active learning, the learner is given the ability to execute actions during learning. Hence, the learner can, to a certain extent, control the stream of training data. A key challenge of active learning is to select actions so as to optimize the rate of learning.

This paper reviews and discusses several heuristic approaches to the selection of actions during active learning, which can primarily be found in the literature on neural networks and reinforcement learning. In order to illustrate how these ideas work in practice, a concrete exploration mechanisms based on artificial neural networks is outlined. In this approach, exploration is achieved through estimating the learner's competence. The basic philosophy behind this and most other approaches to the selection of actions is to estimate the expected gain of knowledge as a function of the actions to be executed. The learner, then, picks actions which maximize the expected knowledge gain.

While the area of passive, order-free learning has been studied extensively in the field of machine learning, statistics, and artificial neural networks, considerably little effort has been spent on the exploration of active learning issues. This might be attributed to the fact that passive learning is simpler, and many learning tasks do not provide the opportunity to select actions. Natural learners such as animals and humans, however, learn actively, as they make use of their capability to act and influence their environment. They are truly embedded in their environments, they act *and* observe. If artificial agents need to learn autonomously, they most likely will have to follow the same learning principles.

References

Angluin, D., 1988, Queries and concept learning, Machine Learning, 2(4):319–342.

Atlas, L., Cohn, D., Ladner, R., El-Sharkawi, M. A., Marks, R. J., Aggoune, M. E., and Park, D. C., 1990, Training connectionist networks with queries and selective sampling, Advances in Neural Information Processing Systems 2, (Touretzky, D., Ed.), pp. 567–573, San Mateo, CA, Morgan Kaufmann.

Barto, A. G. and Singh, S. P., 1990, On the computational economics of reinforcement learning, Connectionist Models, Proceedings of the 1990 Summer School, (Touretzky, D. S., Elman, J. L., Sejnowski, T. J., and Hinton, G. E., Eds.), pp. 35–44, San Mateo, CA, Morgan Kaufmann.

Barto, A. G., Bradtke, S. J., and Singh, S. P., to appear, Learning to act using real-time dynamic programming, Artificial Intelligence.

Baum, E. B. and Lang, K. J., 1991, Constructing hidden units using examples and queries, Advances in Neural Information Processing Systems 3, (Lippmann, R. P., Moody, J. E., and Touretzky, D. S., Eds.), pp. 904–910, San Mateo, Morgan Kaufmann.

Cohn, D., 1994, Queries and exploration using optimal experiment design, Advances in Neural Information Processing Systems 6, (Cowan, J.D., Tesauro, G., and Alspector, J., Eds.), San Mateo, CA, Morgan Kaufmann.

Kaelbling, L. P., 1993, Learning in Embedded Systems. MIT Press, Cambridge, MA.

Koenig, S. and Simmons, R. G., 1993, Complexity analysis of real-time reinforcement learning, Proceeding of the Eleventh National Conference on Artificial Intelligence AAAI-93, pp. 99–105, Menlo Park, CA, AAAI, AAAI Press/The MIT Press.

Korf, R. E., 1988, Real-time heuristic search: New results, Proceedings of the sixth National Conference on Artificial Intelligence (AAAI-88), pp. 139–143, Los Angeles, CA 90024, Computer Science Department, University of California, AAAI Press/MIT Press.

Moore, A. W., 1990, Efficient Memory-based Learning for Robot Control, PhD thesis, Trinity Hall, University of Cambridge, England.

Paass, G. and Kindermann, J., 1995, Bayesian query construction for neural network models, Advances in Neural Information Processing Systems 7, San Mateo, CA, Morgan Kaufmann, (to appear).

Schaal, S. and Atkeson, C. G., 1994, Assessing the quality of learned local models, Advances in Neural Information Processing Systems 6, San Mateo, CA, Morgan Kaufmann.

Sutton, R. S., 1990, Integrated architectures for learning, planning, and reacting based on approximating dynamic programming, Proceedings of the Seventh International Conference on Machine Learning, June 1990, pp. 216–224, San Mateo, CA, Morgan Kaufmann.

Thrun, S. B., 1992, Efficient exploration in reinforcement learning, Technical Report CMU-CS-92-102, Carnegie Mellon University, Pittsburgh, PA 15213.

Thrun, S. B., 1992, The role of exploration in learning control, *Handbook of intelligent control: neural, fuzzy and adaptive approaches*, (White, David A. and Sofge, Donald A., Eds.). Van Nostrand Reinhold, Florence, Kentucky 41022.

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

Whitehead, S. D., 1991, Complexity and cooperation in Q-learning, *Proceedings of the Eighth International Workshop on Machine Learning*, (Birnbaum, L.A. and Collins, G.C., Eds.), pp. 363–367, San Mateo, CA, Morgan Kaufmann.