Learning agents for uncertain environments (extended abstract)

Stuart Russell[*]
Computer Science Division
University of California
Berkeley, CA 94720
russell@cs.berkeley.edu

Abstract

This talk proposes a very simple "baseline architecture" for a learning agent that can handle stochastic, partially observable environments. The architecture uses reinforcement learning together with a method for representing temporal processes as graphical models. I will discuss methods for learning the parameters and structure of such representations from sensory inputs, and for computing posterior probabilities. Some open problems remain before we can try out the complete agent; more arise when we consider scaling up.

A second theme of the talk will be whether reinforcement learning can provide a good model of animal and human learning. To answer this question, we must do *inverse reinforcement learning*: given the observed behaviour, what reward signal, if any, is being optimized? This seems to be a very interesting problem for the COLT, UAI, and ML communities, and has been addressed in econometrics under the heading of *structural estimation of Markov decision processes*.

1 Learning in uncertain environments

AI is about the construction of intelligent *agents*, i.e., systems that perceive and act effectively (according to some performance measure) in an environment. I have argued elsewhere Russell and Norvig (1995) that most AI research has focused on environments that are static, deterministic, discrete, and fully observable. What is to be done when, as in the real world, the environment is dynamic, stochastic, continuous, and partially observable?

[*]This paper draws on a variety of research efforts supported by NSF (IRI-9634215), ONR (N00014-97-1-0941), and ARO (DAAH04-96-1-0341).

In recent years, *reinforcement learning* (also called *neurodynamic programming*) has made rapid progress as an approach for building agents automatically (Sutton, 1988; Kaelbling et al., 1996; Bertsekas & Tsitsiklis, 1996). The basic idea is that the performance measure is made available to the agent in the form of a *reward function* specifying the reward for each state that the agent passes through. The performance measure is then the sum of the rewards obtained. For example, when a bumble bee forages, the reward function at each time step might be some combination of the distance flown (weighted negatively) and the nectar ingested.

Reinforcement learning (RL) methods are essentially online algorithmd for solving *Markov decision processes* (MDPs). An MDP is defined by the reward function and a *model*, that is, the state transition probabilities conditioned on each possible action. RL algorithms can be *model-based*, where the agent learns a model, or *model-free*—e.g., Q-learning cite-Watkins:1989, which learns just a function $Q(s,a)$ specifying the long-term value of taking action a in state s and acting optimally thereafter.

Despite their successes, RL methods have been restricted largely to *fully observable* MDPs, in which the sensory input at each state is sufficient to identify the state. Obviously, in the real world, we must often deal with *partially observable* MDPs (POMDPs). Astrom (1965) proved that optimal decisions in POMDPs depend on the *belief state* $\mathbf{b}$ at each point in time, i.e., the posterior probability distribution over all possible actual states, given all evidence to date. The functions V and Q then become functions of $\mathbf{b}$ instead of s. Parr and Russell (1995) describes a very simple POMDP RL algorithm using an explicit representation of $\mathbf{b}$ as a vector of probabilities, and McCallum (1993) shows a way to approximate the belief state using recent percept sequences.

Neither approach is likely to scale up to situations with large numbers of state variables and long-term temporal dependencies. What is needed is a way of representing the model compactly and updating the belief state efficiently given the model and each new observation. *Dynamic Bayesian networks* (Dean & Kanazawa, 1989) seem to have some of the required properties; in particular, they have significant advantages over other approaches such as Kalman filters and hidden Markov models. Our *baseline architecture*, shown in Figure 1, uses DBNs to represent and update the belief state as new sensor information arrives. Given a representation for $\mathbf{b}$, the reward signal is used to learn a Q-function represented by some "black-box" function approximator such as a neural network. Provided we can handle hybrid (dis-

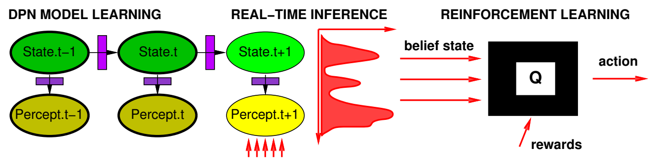

Figure 1: A baseline architecture for learning agents in uncertain environments

crete+continuous) DPNs, and provided we have a learning algorithm that can construct an approximately correct DBN model from scratch, then this baseline architecture has the capacity, *in principle*, to be thrown into more or less *any* environment and to learn to behave reasonably.[1]

The talk will cover a variety of research topics arising from this proposal:

- Parametric learning in DBNs (Binder, Koller, Russell, & Kanazawa, 1997a).
- Structural learning in DBNs (Friedman, Murphy, & Russell, 1998).
- Approximate inference in DBNs (Kanazawa, Koller, & Russell, 1995; Boyen & Koller, 1998).
- Space-efficient inference in DBNs (Binder, Murphy, & Russell, 1997b).
- Reinforcement learning with DBN models—that is, how to do Q-learning with the belief state information provided by the DBN. Some tentative ideas will be presented but as yet there are no convincing solutions.

Scaling up the environment will inevitably overtax the resources of the baseline architecture. There are several obvious directions for improvement, including hierarchical and first-order models, hierarchical representations of behaviour (Parr & Russell, 1998), and model-based lookahead methods for decision making. Which of these is important in any particular class of environments can only be ascertained by experiment.

2 Inverse reinforcement learning

Reinforcement learning is a powerful method for adaptive control in real tasks, so it is natural to seek analogous mechanisms in nature. Connections have been made between reinforcement learning and operant conditioning models of animal learning (see, e.g., Schmajuk & Zanutto, 1997; Touretzky & Saksida, 1997). There is also neurophysiological evidence that reinforcement learning occurs in bee foraging (Montague et al., 1995) and in songbird vocalization (Doya & Sejnowski, 1995).

In this work, it is generally assumed that the reward function is fixed and known. For example, in experiments on bees it is assumed to be the rate of nectar ingestion: Montague et al. (1995) cite evidence of a "neuron with widespread projections to odour processing regions of the honeybee brain whose activity represents the reward value of gustatory stimuli."

It seems clear, however, that *in examining animal and human behaviour we must consider the reward function as an unknown to be ascertained.* The reasons for this are straightforward:

- The specification of a given reward function is an empirical hypothesis and may turn out to be wrong. For example, it was assumed initially that horses' gait selection for a given speed was determined by energetic economy (Hoyt & Taylor, 1981); this turns out not to be the case (Farley & Taylor, 1991).
- The parameters of a *multiattribute* reward function can surely not be determined *a priori*; e.g., for running, attributes might be speed, efficiency, stability against perturbations, wear and tear on muscles, tendons, and bones, etc. How are these to be weighted and combined?

Therefore, to model natural learning using reinforcement learning ideas, we must first solve the following computational task, which we call *inverse reinforcement learning*:

Given 1) measurements of an agent's behaviour over time, in a variety of circumstances, 2) measurements of the sensory inputs to that agent; 3) a model of the physical environment (including the agent's body).

Determine the reward function that the agent is optimizing.

Given an assumption of optimization, this computational task is well-defined. Notice that is the dual of unsupervised reinforcement learning, where the task is to determine optimal behaviour given the reward inputs.

To our knowledge, this computational task has not been studied in any generality in computer science, control theory, psychology, or biology. The closest work is in economics, where the task of *multiattribute utility assessment* has been studied in depth—that is, how does a person actually combine the various attributes of each available choice when making a decision. The theory is well-developed (Keeney & Raiffa, 1976), and the applications numerous. However, this field studies only *one-shot* decisions where a single action is taken and the outcome is immediate. The sequential case was not considered until a seminal paper by Sargent (1978) tried to ascertain the effective hiring cost for labor by examining a firm's hiring behaviour over time, assuming it to be rational. In the last decade, the area of *structural estimation of Markov decision processes* has grown rapidly in econometrics (Rust, 1994). Many of the basic results carry over to our setting, although virtually nothing has been done on computational aspects, experimentation, or control-type applications. The open research problems are many:

- What are efficient algorithms for solving the inverse reinforcement learning problem? What is its computational complexity? Are there closed-form solutions for some parametric forms?
- Under what circumstances can we determine the existence of a consistent reward function? To what extent is the reward function uniquely recoverable?
- What effect do sensor and process noise have on robustness of the determination? What are appropriate error metrics for fitting?

[1] We say "more or less" because full generality require dealing with game-theoretic issues requiring stochastic decision making.

- If behaviour is strongly inconsistent with optimality, can we identify "locally consistent" reward functions for specific regions in state space?
- Can we determine the reward function by observation *during* rather than *after* learning?
- How much observation is required to determine an estimated reward function that is within ϵ of the true reward function?
- How can experiments be designed to maximize the identifiability of the reward function?

Considering the design of possible algorithms, one can take maximum-likelihood approach to fit a parametric form for the reward functionas is commonly done in econometrics. That is, one defines a function $L_r(\mathbf{w})(B)$, the likelihood of observing behaviour B if the true reward function is $r(\mathbf{w})$. From this, one can compute $\partial L/\partial \mathbf{w}$. One important question will be how to compute this gradient efficiently; presumably, it can be done in an obvious way by carrying the differential operator through the optimization algorithm for the behaviour. More elegant closed-form solutions may exist in special cases (e.g., linear-quadratic regulators). One may also be able to show that in some cases (e.g., linear reward functions) a globally optimal estimate can always be found.

The solution of inverse reinforcement learning problems may also be an effective way to learn from observing experts. For tasks such as walking, diving, and driving, the designer of an artificial system may have only an intuitive idea of the appropriate reward function to be supplied to an RL algorithm in order to achieve "desirable" behavior. Instead of learning direct control functions from observation of experts (as in Pomerleau's ALVINN driging system), it may be better to solve the inverse reinforcement learning problem. The reward function should usually be a simple monotonic function of the current sensory inputs, and thus may be *much* simpler than the direct decision mapping itself. That is, the most compact and hence robustly learnable representation of expert behavior may be the reward function.

References

Astrom, K. J. (1965). Optimal control of Markov decision processes with incomplete state estimation. *J. Math. Anal. Applic.*, *10*, 174–205.

Bertsekas, D. C., & Tsitsiklis, J. N. (1996). *Neuro-dynamic programming*. Athena Scientific, Belmont, Mass.

Binder, J., Koller, D., Russell, S., & Kanazawa, K. (1997a). Adaptive probabilistic networks with hidden variables. *Machine Learning*, *29*, 213–244.

Binder, J., Murphy, K., & Russell, S. (1997b). Space-efficient inference in dynamic probabilistic networks. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97)* Nagoya, Japan. Morgan Kaufmann.

Boyen, X., & Koller, D. (1998). Tractable inference for complex stochastic processes. In *Proc. 14th Annual Conference on Uncertainty in AI (UAI)*. to appear.

Dean, T., & Kanazawa, K. (1989). A model for reasoning about persistence and causation. *Computational Intelligence*, *5*(3), 142–150.

Doya, K., & Sejnowski, T. (1995). A novel reinforcement model of birdsong vocalization learning. In Tesauro, G., Touretzky, D., & Leen, T. (Eds.), *Advances in Neural Information Processing Systems*, Vol. 8, pp. 101–8 Denver, CO. MIT Press.

Farley, C. T., & Taylor, C. R. (1991). A mechanical trigger for the trot–gallop transition in horses. *Science*, *253*(5017), 306–308.

Friedman, N., Murphy, K., & Russell, S. (1998). Learning the structure of dynamic probabilistic networks. In *Uncertainty in Artificial Intelligence: Proceedings of the Fourteenth Conference* Madison, Wisconsin. Morgan Kaufmann.

Hoyt, D., & Taylor, C. (1981). Gait and the energetics of locomotion in horses. *Nature*, *292*, 239–240.

Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, *4*, 237–285.

Kanazawa, K., Koller, D., & Russell, S. (1995). Stochastic simulation algorithms for dynamic probabilistic networks. In *Uncertainty in Artificial Intelligence: Proceedings of the Eleventh Conference*, pp. 346–351 Montreal, Canada. Morgan Kaufmann.

Keeney, R. L., & Raiffa, H. (1976). *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. Wiley, New York.

McCallum, A. R. (1993). Overcoming incomplete perception with utile distinction memory. In *Proceedings of the Tenth International Conference on Machine Learning*, pp. 190–196 Amherst, Massachusetts. Morgan Kaufmann.

Montague, P. R., Dayan, P., Person, C., & Sejnowski, T. J. (1995). Bee foraging in uncertain environments using predictive hebbian learning. *Nature*, *377*, 725–728.

Parr, R., & Russell, S. (1995). Approximating optimal policies for partially observable stochastic domains. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95)* Montreal, Canada. Morgan Kaufmann.

Parr, R., & Russell, S. (1998). Reinforcement learning with hierarchies of machines. In Kearns, M. (Ed.), *Advances in Neural Information Processing Systems 10*. MIT Press, Cambridge, Massachusetts.

Russell, S. J., & Norvig, P. (1995). *Artificial Intelligence: A Modern Approach*. Prentice-Hall, Englewood Cliffs, New Jersey.

Rust, J. (1994). Do people behave according to bellman's principal of optimality?. Submitted to Journal of Economic Perspectives.

Sargent, T. J. (1978). Estimation of dynamic labor demand schedules under rational expectations. *Journal of Political Economy*, *86*(6), 1009–1044.

Schmajuk, N. A., & Zanutto, B. S. (1997). Escape, avoidance, and imitation: a neural network approach. *Adaptive Behavior*, *6*(1), 63–129.

Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Machine Learning*, *3*, 9–44.

Touretzky, D. S., & Saksida, L. M. (1997). Operant conditioning in Skinnerbots. *Adaptive Behavior*, *5*(3–4), 219–47.