CS497 Lecture Notes - Feb. 25, Feb. 27

Scribed by Michael Treaster

March 7, 2003

Reinforcement Learning

Reinforcement learning helps an decision-maker (DM) to make decisions by exploring offline the state/decision space to figure out on its own the results of a given action at a given state. ($\mathrm{P}(x'|x,u)$) This offline learning relies on a having a simulation of environment in which the DM is going to act, or on being able to spend a significant amount of time making mistakes in the real problem space. Reinforcement learning blends the ideas of dynamic programming, Robbins-Monro, and stochastic iterative algorithms. The most significant issue in reinforcement learning is the balance of taking irrational actions to explore in new directions and potentially find a better solution versus exploiting acquired knowledge to ensure exploring a reasonable region of the state/decision space.

In a given state, the DM chooses an action, u. Then nature chooses an action, θ, resulting in a new state. The DM can make observation-based estimates of what will happen when a particular action is selected in a particular state based on this movement through the state space.

The update equation is:

$$\hat{J}_\gamma(x) := (1-\rho) \underbrace{\hat{J}_\gamma(x)}_{\text{prev. estimate}} + \rho(\underbrace{\ell(x,\gamma(x))}_{\text{Loss at } x} + \underbrace{\alpha \hat{J}_\gamma(x')}_{\text{Est. cost-to-go for } x'})$$

This equation shows how $\hat{J}_\gamma(x)$ learns the cost-to-go for a state x. With each iteration, the estimated cost-to-go for state x is updated by combining the previous estimate for x with the cost of the action u taken at x and the estimated cost-to-go for the resulting state x'.

ρ is the **learning rate** $\in (0,1)$, with typical $\rho \in (0.01, 0.5)$. This represents how much influence new observations have on what has already been learned. A high ρ will cause the $J_\gamma(x)$ to fluctuate wildly, while a small ρ will result in slow or nonexistent learning performance.

Q-Learning

Q-Learning is an example of a reinforcement learning. It is guaranteed to find an optimal strategy given a sufficient number of training iterations.

Previously $J^* : X \to \Re$ optimal cost to go

Q-learning $Q^* : X \times U \to \Re$ action dependent cost to go

The Q-function is not specialized to Q-learning. In fact, it is a more general method than the J-functions we have used in the past. We could have used Q-functions in place of J-functions, but we would have come to the same result after more work.

We use the following equation to obtain a dynamic programming solution. Q^* represents the expected cost of the move sequence using the optimal strategy starting at x. This assumes that all moves except the current one are optimal.

$$Q^*(x,u) = \ell(x,u) + \sum_{x'} (P(x'|x,u)\alpha \min_{u' \in U(x')} (Q^*(x',u'))$$

Based on the above equation, we can obtain the following update equation. Repeatedly evaluating this equation for a variety of (x,u) will eventually result in an optimal solution. Certain conditions are required to guarantee convergence in different situations. Also, this equation assumes simple, finite problems, but this is not a requirement of Q-learning.

$$\hat{Q}^*(x,u) := (1-\rho)\hat{Q} * (x,u) + \rho(\ell(x,u) + \alpha \min_{u \in U(x')} \hat{Q}^*(x',u'))$$

Imperfect State Information

In reality, we are usually working with imperfect state information, especially in the area of robotics. Unfortunately, it is easy to have this situation explode into an infinite-dimensional space. Keep in mind that the goal in this situation is to make good decisions. Although it is helpful to know the current state, it is not actually the goal.

Suppose we are in an unknown state, x_k, but we are able to make an observation/measurement/sensor reading y_k which contains information about x_k. Nature is involved as well. As in decision theory, it can interfere with actions when it chooses its move, θ_k. However, it can also interfere with observations.

ϕ_k = a nature observation action

Φ = the set of observation actions

$y_k = h(x_k, \phi_k)$ = the observation equation

The **observation equation**

$$y_k = h(x_k, \phi_k)$$

says that some observation, y_k, is a function of the state and nature's interference. Of course, since we have imperfect information we don't know what x_k is, but we can still use it in this model even if we don't know what all the pieces are.

There are two kinds of uncertainty:

Projection: Occurs without nature's interference, and results from incomplete state information. The observation equation is $y_k = h(x_k)$. For example, perhaps the DM is exploring an (x,y) space but can only observe the x-coordinate.

Disturbance: Nature is a nuisance, giving incorrect state information. For example, a GPS receiver returns only approximate values for x, y, and z coordinates.

The initial condition for the sequence of decisions could be one of several types of situations, depending on the information that is available.

Perfect Information: The initial state x_1 is given

Non-deterministic: A set of possible initial states $X_1 \subseteq X$ is given

Probabilistic: The probability of beginning in a given state, $P(x_1)$, is given

In addition to the initial condition, we also have perfect information on the series of actions and perfect information on the series of observations received. Thus, at any k we have the **information state** (or history) η_k on which to base decisions.

$$\eta_k = \{IC, u_1, u_2, \ldots, u_{k-1}, y_1, y_2, \ldots, y_k\}$$

N_k denotes the **information space**, which is the set of all information states. Since an information state can hold a sequence of moves and observations of potentially infinite length, N_k has infinite dimension in the limit as $k \to \infty$.

In a situation with perfect state information, a decision can be made using strategy $\gamma : X \to U$ to find $u = \gamma(x)$. In a situation with imperfect state information, the strategy must intead be defined in terms of the information space. Thus, the strategy $\gamma_k : N_k \to U$ will allow us to find $u_k = \gamma(\eta_k)$. Since we are unable to know the state with perfect certainty, we may not know what moves are available at any given time. To solve this problem, we will assume that all actions are available at all times.

N thus becomes a "super state space", in which the DM has perfect information. In this space of perfect information, we could fall back on classical decision theory to make decisions. In practice, however, this state space is too large. Data must be culled from the information state to reduce the problem to a tractable form.

Manipulating the Information Space

Limited memory: Only remember the last i stages of actions and observations. This is not always a sensible or feasible solution, since early information can sometimes be critical to determining the current state.

Approximate N_k

Find equivalence classes in N_k: Find symmetries in the information space to reduce the number of dimensions or states that need to be stored.

Be Bayesian: Use Bayes' rule to produce a probability distribution for X. Instead of carrying around the infinite-dimensional N to resolve decisions. Instead we can store only a probability distribution of what the DM believes the current state to be. This will have dimension equal to the number of states in X.

$$\begin{aligned} \text{Apply } u_k \quad & P(x_{k+1}|\eta_k, u_k) = \sum_{x_k} P(x_{k+1}|x_k, u_k) P(x_k|\eta_k) \\ \text{Apply } y_{k+1} \quad & P(x_{k+1}|\eta_k, u_k, y_{k+1}) = \frac{P(y_{k+1}|x_{k+1}) P(x_{k+1}|\eta_k, y_k)}{\sum_{x_{k+1}} P(y_{k+1}|x_{k+1}) P(x_{k+1}|\eta_k, y_k)} \end{aligned}$$

Nondeterministic "Bayesian": Infer the set of state possibilities and, based on this set of possibilities the DM can assess the possible outcomes of possible actions and act accordingly.

$$\begin{aligned} \text{Apply } u_k \quad & S_{k+1}(u_k, \eta_k) = \cup_{\theta_k \in \Theta} \{ \cup_{x_k \in S_k(\eta_k)} f(x_k, u_k, \theta_k) \} \\ \text{Apply } y_{k+1} \quad & S_{k+1}(\eta_k) = S_{k+1}(u_k, \eta_k) \cap S_{k+1}(y_{k+1}) \end{aligned}$$

In these equations, the $\cup$ operation serves the same purpose as the marginalization from the Bayesian version, increasing uncertaintly. The $\cap$ operation matches the possible destination states with the possible observation states to find the possible current states, thus reducing uncertainty.

All of these methods assume that if two histories lead to the same probability distributions, sets, or other state representation that the two histories mathematically indistinguishable for purposes of decision making.

Possible Goals

Although imperfect information decision making can often be motivated by the same sorts of goals as perfect information decision making, it can also be motivated by goals as well. For example, the goal might be to reach a particular state with a particular degree of certainty, or to know with a particular degree of certainty what state the DM is in, regardless of what state that happens to be.