1 Reinforcement Learning (continued)

1.1 Stochastic Iterative Algorithm: Robbins-Monro

Reinforcement learning is the idea of blending the three stages of optimization (learning, planning, and execution) into one operation. The biggest issue when doing so is the consideration of Exploration vs. Exploitation:

**Exploration** Try to gather information about $P(x' | x, u)$, the probabilities inherent in the environment. The problem here is that in the process of gathering the learner may have to choose actions that may have high losses.

**Exploitation** Make good decisions based on the current knowledge of $P(x' | x, u)$. In this case the danger is that the learner may fail to find a better solution due to lack of learning, the “can’t teach a dog new tricks” syndrome.

So in reinforcement learning the decision planning equations will be modified to allow learning to occur. First let’s focus on Temporal Difference Learning TD(0), where the strategy is updated incrementally at every stage. Using the Robbins-Monro iterations on a stochastic iterative algorithm $y = h(y)$, we solve $y = h(y)$ with noisy observations of $h(y)$.

$$y := (1 - \rho) y + \rho(h(y))$$

$\rho \in (0, 1)$

Recall the expected loss for a given strategy:

$$J_\gamma(x) = l(x, \gamma(x)) + \alpha \sum_{x'} P(x' | x, u) J_\gamma(x'), \ u = \gamma(x), \ x' = f(x, u)$$

Combining ideas, we then have

$$y = J_\gamma(x)$$

$$h(y) = l(x, \gamma(x)) + \alpha J_\gamma(x')$$

$$J_\gamma(x) := (1 - \rho) J_\gamma(x) + \rho(l(x, \gamma(x))) + \alpha J_\gamma(x')$$

where $x'$ is the observed state and $\rho \in (0, 1)$ is the **learning rate** for TD(0). Typically $\rho \in [0.01, 0.5]$. If $\rho$ is too small, then there isn’t a sufficiently high rate of learning; if, on the other hand, $\rho$ is too large, then there is the danger of $J_\gamma(x)$ wildly fluctuating rather than converging. With a properly sized value, and if $\rho$ is gradually decreased over time (i.e. the estimation is gradually fine-tuned in smaller increments), then $J_\gamma(x)$ should properly converge to the expected loss of $\gamma$, $J_\gamma(x)$. 
1.2 Finding an Optimal Strategy: Q-learning

So how do we find the optimal strategy? The answer lies in $Q$: rather than using just $J^* : X \to \mathbb{R}$, the expected loss of a particular strategy, now we use $Q^* : X \times U \to \mathbb{R}$. $Q^*(x, u)$ represents the optimal cost-to-go from applying $u$ and then continuing on the optimal path after that. Note that $Q$ is independent of the policy being followed.

Using $Q^*(x, u)$ in the dynamic programming equation yields:

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

If we make $J^*(x)$ the expected cost for optimal strategy given state $x$, and $Q^*(x, u)$ be the expected cost for optimal strategy given state $x$ and using cost $u$, then

$$J^*(x) = \min_{u \in U(x)} Q^*(x, u)$$

However, for reinforcement learning, the probability $P(x'|x, u)$ is unknown, so we can bring in the stochastic iterative idea again and get

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

2 Imperfect State Information

Suppose $x^k$ is unknown. We have an observation (sometimes called a measurement or sensor reading) $y_k$ which contains information about $x_k$.

Let nature interfere with the observations.

Let $\phi_k$ denote a nature observation action.

Let $\Phi$ denote the set of nature observation actions.

Let $y_k = h(h_k, \phi_k)$ be the observation equation.

There are two kinds of uncertainties:

**Projection** Even if $y_k = h(x_k)$, $y_k$ could have a lower dimension than $x_k$ and thus possibly not take into account all the variables. $y_k$ would thus be like a feature vector that “approximates” $x_k$.

**Disturbance** Since $\phi_k$ is unknown, it can be considered as a disturbance applied to the observation as $y_k = x_k + \phi_k$, a sort of “jittering” from nature.

Now imagine trying to make a decision. What information is available?

Initial Conditions (referred to as I.C.)

1. $x$ is given, or
2. $X_1 \subseteq X$ (non-deterministic), or
3. $P(x_1)$ (probabilistic)

Observation History: $y_1, y_2, \ldots, y_k$

Action History: $u_1, u_2, \ldots, u_{k-1}$

Decisions can now be based on an information state (history):

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

Let $N_k$ denote the set of all information states for stage $k$, the information space. Note that $\eta_k$ contains all of the information that could possibly used to influence a decision. What, then, is the dimension of $N_k$?

$$\dim(N_k) = k \cdot \dim(Y) + (k - 1) \cdot \dim(U) + \dim(\text{set of all I.C.'s})$$
Consider that 
\[ \lim_{k \to \infty} \dim(N_k) = \infty \]
and it’s easy to see that \(N_k\) can grow to be enormous.

Now, for defining a strategy:

**Perfect state information** \( \gamma : X \to U \), \( u = \gamma(x) \) i.e. strategies only need the current state to choose the next action.

**Imperfect state information** \( \gamma_k : N_k \to U \), \( u_k = \gamma(\eta_k) \in U \) i.e. the next action depends on the history of the actions and states previously visited.

Note that it is difficult to know which actions are available; \( U(x) \) is unknown because \( x \) is unknown. Assume here that \( U(x) = U \) for all \( x \in X \).

### 2.1 Manipulating the Information Space

The iterative optimization process is graphically represented in the following manner:

Starting at state \( x_1 \), receive observation \( y_1 \) and then decide on an action \( u_1 \) to reach state \( x_2 \). Then receive observation \( y_2 \), decide on action \( u_2 \) to reach a state \( x_3 \), and so on.

Possible solutions to preventing the information space from growing arbitrarily through the dimensions being too high:

1. Limiting memory
2. Be Bayesian — \( P(x_k|\eta_k) \)
3. Be nondeterministic — \( S_k(\eta_k) \)
4. Approximate \( N_k \)
5. Find equivalence classes in \( N_k \)

First, a recap of the variables involved:

- \( y_k = h(x_k, \phi_k) \)
- \( x_{k+1} = f(x_k, u_k, \theta_k) \)
- \( \eta_k = \{ I.C., u_1, u_2, \ldots u_{k-1}, y_1, y_2, \ldots y_k \} \)
- \( N_k = \) information space
- \( \gamma : N_k \to U \) (not \( \gamma : Y \to U \), which only would consider the current state)

#### 2.1.1 Limited Memory

With limited memory only the last \( i \) stages would be remembered, so \( \eta_k = \{ u_{k-1}, \ldots, u_{k-1}, y_{k-i+1}, \ldots, y_k \} \). If \( i = 1 \), then the information state would simply be \( \eta_k = \{ u_{k-1}, y_k \} \) or even just \( \eta_k = \{ y_k \} \) for the last observation on the current state, ignoring the last action. Of course, the disadvantage here is that this gives the learner a short-term memory, and the early information that wasn’t saved could be important to the current decision. If, for instance, the situation was a duel with guns, and the agreement was to take twenty steps before turning and firing, someone who only had enough memory to count fifteen steps would be in trouble...
2.1.2 Bayesian

Now we consider a probabilistic approach. We first eliminate nature from the equations:

\[ P(\theta_k|x_k, u_k) \text{ given } \sim \Rightarrow P(x_{k+1}|x_k, u_k) \text{ since } x_{k+1} = f(x_k, u_k, \theta_k) \]

\[ P(\phi_k|x_k) \text{ given } \sim \Rightarrow P(y_k|x_k) \text{ since } y_k = h(x_k, \phi_k) \]

Assume \( P(x_1) \) for the Initial Condition.

First, receive observation \( y_1 \). Use Bayes Rule to obtain

\[ P(x_1|y_1) = \frac{P(y_1|x_1)P(x_1)}{\sum_{x_1}P(y_1|x_1)P(x_1)} \]

What do we have now?

\[ P(x_2|x_1, u_1) \text{ from } f \text{ and } P(\theta) \]

\[ P(x_1|y_1) \text{ from Bayes Rule} \]

Apply \( u_1 \) to obtain \( x_2 \):

\[ P(x_2|u_1, y_1) = \sum_{x \in X} P(x_2|x_1, u_1)P(x_1|y_1) \]

Note that \( y_1 \) is missing from \( P(x_2|x_1, u_1) \), since \( y_1 \) is conditionally independent from \( x_2 \) (no need for a previous observation if the action’s already been determined). Also, \( u_1 \) is missing from \( P(x_1|y_1) \) because \( u_1 \) has no bearing on \( x_1 \), being the action to determine \( x_2 \).

For the more general case, use induction: given \( P(x_k|\eta_k) \), determine \( P(x_{k+1}|\eta_{k+1}) \):

\[ \eta_{k+1} = \eta_k \cup \{u_k, y_{k+1}\} \]

First handle \( u_k \):

\[ P(x_{k+1}|u_k) = \sum_{x \in X} P(x_{k+1}|x_k, u_k)P(x_k|\eta_k) \]

Now handle \( y_{k+1} \):

\[ P(x_{k+1}|\eta_k, u_k, y_{k+1}) = \frac{P(y_{k+1}|x_{k+1})P(x_{k+1}|\eta_k, u_k)}{\sum_{x_{k+1}} P(y_{k+1}|x_{k+1})P(x_{k+1}|\eta_k, u_k)} \]

This means that an information state for Markov Decision Processes can be viewed as a probability distribution

\[ \eta_k \rightarrow P(x_k|\eta_k) \]

It’s possible that two different information spaces lead to the same probability, i.e. \( P(x_k|\eta_k) = P(x_k|\eta'_k) \). However, even if the histories are different, since the probabilities are the same, then information state for an MDP is essentially the same.

2.1.3 Nondeterministic

\( X_1 \subseteq X \) is the set of possible initial states.

Let \( S_k(\sim) \subseteq X \) denote the set of possible states given the information in \( \sim \), where \( \sim \) is the same conditions given in a Bayesian probability \( P(x_k|\sim) \).

Here we’re assuming \( \Phi \), the set of observable actions by nature, is given. From this we obtain \( S_k(y_k) \subseteq X \):

\[ S_k(y_k) = \{x_k \in X \mid \exists \phi_k \in \Phi \text{ for which } x_k = h(y_k, \phi_k)\} \]

Starting with the Initial Condition \( S_1(I.C.) = X_1 \), observe \( y_1 \):

\[ S_1(I.C., y_1) = S_1(\eta_1) = S(\eta_1) \cap X_1 \]
Now, choose $u_1$:

$$S_2(u_1, \eta_1) = \{ x_2 \in X \mid \exists \theta_1 \in \Theta, \exists x_1 \in S_1(\eta_1), \text{ for which } x_2 = f(x_1, u_1, \theta_1) \}$$

Alternately, this can also be represented as

$$S_2(u_1, \eta_1) = \bigcup_{\theta_1 \in \Theta} \bigcup_{x_1 \in S_1(\eta_1)} f(x_1, u_1, \theta_1)$$

That is, $S_2$ is equal to the union of all possible states reachable by applying action $u_1$ and nature’s influence $\theta_1$ to all the states in $S_1(\eta_1)$. Thus, an infinite number of information states is reduced to a number of sets for any given stage, i.e. $S_1(\eta_1), S_2(\eta_2), S_3(\eta_3) \ldots$ and so on.

Now, if we observe $y_2$:

$$S_2(u_1, y_2, \eta_1) = S(\eta_2) = S_2(u_1, \eta_1) \cap S_2(y_2)$$

where $S_2(y_2)$ is derived from the $S_k(y_k)$ formula from before.

In general, assuming $S_k(\eta_k)$ is given, how do we find $S_{k+1}(\eta_{k+1})$?

Use $S_k(\eta_k)$ to obtain $S_{k+1}(u_k, \eta_k)$ after choosing $u_k$, as above:

$$S_{k+1}(u_k, \eta_k) = \bigcup_{\theta_k \in \Theta} \bigcup_{x_k \in S_k(\eta_k)} f(x_k, u_k, \theta_k)$$

Receive $y_{k+1}$:

$$S_{k+1}(\eta_k, u_k, y_{k+1}) = S_{k+1}(\eta_{k+1}) = S_{k+1}(u_k, \eta_k) \cap S_{k+1}(y_{k+1})$$