

Multi-criteria Reinforcement Learning

Eric Meisner

October 24, 2006

Problem

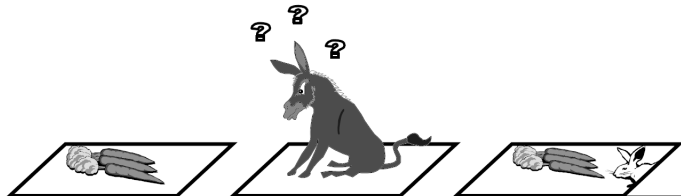
A general Learning problem:

- Given an agent in a world with a finite set of states $x \in X$,
- Agent has a finite set of actions A in each state,
- Each action provides some reward
- Global value of a state is estimated by taking actions and observing rewards

Determine the policy $\pi(x): X \rightarrow A$ which maximizes the expected total reward

Simple Multi-Objective Learning Example: Leibniz's Mule

- Mule can be at Left, Right or Middle
- Food appears and is stolen from Left and Right according to some rule
- If Mule is in the Middle then food can appear but not disappear from Left and Right
- Goals are to Eat as much as possible, prevent as much theft as possible.



Vector valued Rewards

Recall Mr. Nelson's Rain problem:

- There are two states of nature: raining or not raining
- Mr. Nelson must decide what to wear
- For any strategy, there is a loss of utility for each state of nature

Strategies are represented by points in the space of two expected losses

Abstract Dynamic Program

ADP: (R, X, A, A, Q)

- R : return space
- X : state space
- A : set of actions
- $A(x) : X \rightarrow A$ feasible actions for a given state
- $Q : R^X \rightarrow R^{X \times A}$ the reinforcement operator

Evaluation

The value of a state $x \in X$ is the expected total reward when the mule starts at x and uses a particular policy π . This value can be defined recursively as:

$$v_{\pi,1}(x) = E\left(\sum_{t=0}^{\infty} R_t\right)$$

$$v_{\pi,2}(x) = E\left(\sum_{t=0}^{\infty} S_t\right)$$

Weighted Criterion

$$\sum_{t=0}^{\infty} w_1(1 - S_t) + w_2 R_t$$

for $w_1, w_2 > 0$

However such weights may not exist

Goal Programming

For one of the criteria, set a hard constraint, and minimize the other. This gives a typical optimization problem of the form

$$\begin{aligned} & \text{minimize : } f_0(x) \\ & \text{subject to : } f_i(x), i = 1 \dots n \end{aligned}$$

Applying Goal Programming to Learning

Using the Goal Programming approach, set a minimum amount of food per unit time R_{crit} that must be eaten by the mule in order to survive. The new value function is as follows:

$$v_{\pi,1}(x) = \min(R_{crit}, E[\sum_{t=0}^{\infty} R_t])$$

$$v_{\pi,2}(x) = E[\sum_{t=0}^{\infty} S_t]$$

Comparing State Reward Vectors

Use a **reverse-2nd lexicographic ordering**:

$$\begin{aligned} v_{\pi}(x_1) < v_{\pi}(x_2) &\Rightarrow \\ v_{\pi,1}(x_1) < v_{\pi,1}(x_2) &\text{ or if } v_{\pi,1}(x_1) = v_{\pi,1}(x_2) \\ \text{then } v_{\pi,2}(x_1) > v_{\pi,2}(x_2) \end{aligned}$$

And Similarly for comparing policies.

Recursive Form

Removing the expectation operator and using the transition probabilities we have,

$$\begin{aligned}v_{\pi,1}(x) &= \min(R_{crit}, R(x, \pi(x))) \\ &\quad + \min(R_{crit}, \gamma \sum_{y \in X} p(x, \pi(x), y) v_{\pi,1}(y)) \\ v_{\pi,2}(x) &= \sum_{y \in X} p(x, \pi(x), y) [S(x, \pi(x), y) v_{\pi,2}(y)]\end{aligned}$$

Supremum

- Recall from lecture, comparison of policies for Markov Decision Processes
- We have defined an ordering on state reward vectors, so we have a partial ordering of policies.
- Also, recall the existence of optimal policies for MDP's

Optimal Policies for Leibniz's Mule

- For a given R_{crit} the optimal solution is to stay in the middle until eating is necessary.
- Either select a mixed strategy (transitions are probabilistic), or augment state space to include counting up to R_{crit}

The Recursive Value Estimator

- $v_\pi(x)$ is a function which, given a policy π , maps states to real values, or in the multi-criteria case, maps states to R^n .
- $Qv(x, a)$ maps state action pairs to values, is defined similar to $v()$, but parameterizes the action.

$$Qv_1(x, a) = \min(R_{crit}, R(x, a) + \min(R_{crit}, \gamma \sum_{y \in X} p(x, a, y) v_1(y)))$$

$$Qv_2(x, a) = \sum_{y \in X} p(x, a, y) S(x, a, y) v_2(y)$$

The Bellman Optimality Equation

- $T_\pi v(x) = Qv(x, \pi(x))$ is a recursive estimator of the value function of a policy, π . T includes the reinforcement dynamics, and has fixed point v_π as long as X and A are finite.
- $Tv(x) = \text{m. a. x.}_{a \in A(x)} Qv(x, a)$ is the Bellman optimal equation which describes the recursion for expected rewards.

Iterative Solution

The value function and optimal actions can be computed iteratively by:

$$A_{i+1} = a \in A_i(x) \mid \arg \max_a Qv(x, a)_i$$
$$T_\pi v(x)_{i+1} = \arg \max_{a \in A(x)} Qv(x, a)_{i+1}$$

for $i = 1, 2, \dots, n$

Approach

- Show that optimal policies exist
- Show that T has a fixed point
- Show that T is a contraction in order to show that learning is tractable

Convergence

- A sequence a_n **converges** to a real number a if, for every positive number ϵ , there exists an $N \in \mathbf{N}$ such that whenever $n \geq \mathbf{N}$ $|a_n - a| \leq \epsilon$
- A function f , from a metric space (M, d) to itself is a **contraction** if

$$d(f(x), f(y)) \leq k * d(x, y)$$

for $0 < k \leq 1$

The Strategy

Show that the components of Q are contractions:

- T_1 gives the recursive estimates of the first component R and is a contraction and well defined.
- T_2 is contingent on T_1

The Moral

- We have imposed an ordering on vectors in the reward space
- There are optimal stationary policies for the watchman's compromise construction.
- Component-wise analysis of the Q function promises convergence for learning problems.

Wait a minute...

What about the learning part?

- Traditional RL learning algorithms will work if T is a contraction
- The rest of the paper shows a version of Q-learning for multi-criteria w/ m. a. x. which converges if initial estimate of $Q_V(x)$ is an overestimate.
- Simulation results are provided

Questions?

■ ...

Mannor & Shimkin: A Steering Approach

- A learning agent P1 and an arbitrary adversary P2
- Finite state space/actions
- State is a (probabilistic) function of P1 and P2 actions
- At each round P1 gets a vector reward

Goal of P1

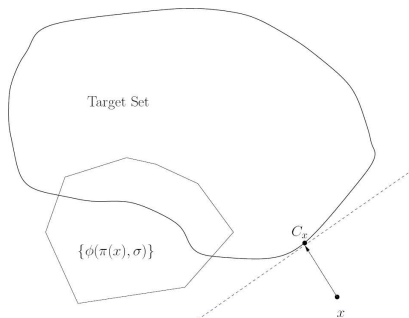
- There is a target set in the reward space
- The game does not terminate
- P1 wants to drive the average reward vector to the target set

Assumptions

- The target set T is **approachable** by P1
- The game is ergodic, so there is a reference state s^* for which the expected return time is finite, for any policies played by P1 and P2
- T is a closed convex set

Approachability

A set T is **approachable** by P1 if for every point $x \in T$, regardless of the policy of P2, P1 can ensure that the per cycle reward vector will be on the other side of the hyper-plane separating x and its closest point in T .



Recurrence

- An MDP combined with a policy yields a *process* which is *ergodic* if it has a well defined stationary distribution.
- The reference state s^* exists if the process is ergodic.

Basic Idea

- The game is played in the reward space
- The average reward is represented by a point in the reward space
- P1 learns to steer the average reward vector in order to keep it in the target set

Steering

- First assume that P1 knows the dynamics of the system
- P1 has a discrete set of directions $U_1, U_2, U_3, \dots, U_j$ and a set of policies $\pi_1, \pi_2, \pi_3, \dots, \pi_j$ for steering in each direction (Follows from approachability)
- P1 can approximately approach T by steering in one of the discrete directions.

Algorithm

- When in the reference state s^* , examine the average reward vector
- Select the direction vector u_x from the current average reward vector x to the closest point in \mathcal{T}
- Select from a finite set of steering directions, the closest vector to u_x and the corresponding policy.

Wait, Forgot About Learning Again...

- Previously assumed the policies $\pi_1, \pi_2, \pi_3, \dots, \pi_j$ were known.
- The steering policies π_j can be learned because the reward is just the Euclidean distance from the final reward vector to the goal vector u_x
- Recall, that the tractability of learning is contingent upon a well defined norm.

Quick Analysis

- Arguably more elegant than watchman's compromise
- Can deal with moving target sets
- Still requires definition of a convex target set.
- Requires several learning agents (more time and space)

References

- Z. Gabor and Z. Kalmar and C. Szepesvari "Multi-criteria reinforcement learning", "Gabor, Z., Kalmar, Z., Szepesvari, C., Multi-criteria reinforcement learning, *International Conference on Machine Learning, Madison* July 1998.
- Shie Mannor and Nahum Shimkin "A Geometric Approach to Multi-Criterion Reinforcement Learning" *The Journal of Machine Learning Research* Volume 5 , (December 2004) table of contents Pages: 325 - 360 Year of Publication: 2004 ISSN:1533-7928
- Michael Kearns and Satinder Singh "Near-optimal reinforcement learning in polynomial time", *Proc. 15th International Conf. on Machine Learning* pages 260-268, 1998.