EE556 Stochastic Systems and Reinforcement Learning
Exam Review - Chapters 2-5 of Jain's "Notes"

2 Stochastic System Models

2.1 Introduction

stochastic system

discrete state space X

$$x_{k+1} = f_k(x_k, u_k, w_k), \quad k = 0, 1, 2, \cdots$$

control action space U.

where at time k

$x_k \in X$ is the state,

$u_k \in U$ is the control action

w_k is the randomness the drives the dynamics, often called plant noise.

x_0 the initial state.

observation of the system is available given by

$$y_k = g_k(x_k, v_k), \quad k = 0, 1, 2, \cdots$$

We note that both state and observation equations may change with time, hence the indexing by k. In addition to $\{(f_k, g_k), k = 0, 1, 2, \cdots\}$, we must also specify the probability distribution of the random variables $x_0, w_0, w_1, \cdots, v_0, v_1, \cdots$, and the probability space $(\Omega, \mathcal{F}, \mathbb{P})$ on which they are defined, to complete the specification of a stochastic system. Unless otherwise stated, we will assume these random variables to be *independent*, and for random variables $(w_k, k \geq 0)$ to be *identically distributed* and similarly for random variables $(v_k, k \geq 0)$.

feedback law $\pi = (\pi_0, \pi_1, \cdots)$

$$u_k = \pi_k(y_k)$$

Control policies are of two types. When a control action sequence is fixed a priori, say $\bar{u}_0, \bar{u}_1, \cdots$, we call it *open-loop control*. The resulting state and observation sequences will be denoted $x_0, \bar{x}_1, \bar{x}_2, \cdots$ and $y_0, \bar{y}_1, \bar{y}_2, \cdots$ respectively. When the control actions are determined as per a feedback law as in (2.3), we call it *closed-loop control*. A key question is which is better, and whether a particular

2.2 Markov Decision Processes

Let X and U be both finite.

transition kernel

$$x_{k+1} \sim P(\cdot | x_k, u_k)$$

conditional probabilities

$$y_k \sim O(\cdot | x_k),$$

observation kernel

FACT 2.2 Under any feedback policy π,

$$P^\pi(X_{k+1} | x^k, u^k) = P^\pi(X_{k+1} | x_k, u_k).$$

FACT 2.3 Under any open-loop policy $\bar{\pi} = (\bar{u}_0, \bar{u}_1, \cdots)$,

$$P^{\pi}(X_{k+m+1}|x^k, \bar{u}^k, \bar{u}_{k+1}^{k+m}) = P^{\pi}(X_{k+m+1}|x_k, \bar{u}_k, \bar{u}_{k+1}^{k+m}).$$

Markov policy

$$\pi = (\pi_0, \pi_1, \cdots)$$

if π_k is a function of x_k alone.

stationary policy

$$\pi_k = \pi$$

i.e., it does not change with time.

finite-horizon Markov Decision Process (MDP).

$$x_{k+1} \sim P(\cdot|x_k, u_k), \quad \text{and} \quad y_k \sim O(\cdot|x_k),$$

`with` objective function

aka "Reward-to-go"

$$J^{\pi} = \mathbb{E}[\sum_{k=1}^{K} r(x_k, u_k)]$$

infinite-horizon discounted-reward Markov Decision Process (MDP).

$$x_{k+1} \sim P(\cdot|x_k, u_k), \quad \text{and} \quad y_k \sim O(\cdot|x_k),$$

$$J^{\pi} = \mathbb{E}[\sum_{k=0}^{\infty} \gamma^k r(x_k, u_k)],$$

infinite-horizon average-reward Markov Decision Process (MDP).

$$x_{k+1} \sim P(\cdot|x_k, u_k), \quad \text{and} \quad y_k \sim O(\cdot|x_k),$$

$$J^{\pi} = \liminf_{K\to\infty} \mathbb{E}[\frac{1}{K} \sum_{k=1}^{K} r(x_k, u_k)],$$

Value Function

$$V_0^\pi(i) := r(x_0,u_0) + \mathbb{E}[\sum_{k=1}^{K} r_k(x_k,u_k)|x_0 = i]$$

$$V_k^\pi(i) = r_k(i,\pi_k(i)) + \mathbb{E}[V_{k+1}(x_{k+1})|x_k = i]$$
$$= r_k(i,\pi_k(i)) + \sum_j P_{ij}^\pi V_{k+1}(j)$$

in matrix notation

$$V_k^\pi = r_k^\pi + P_k^\pi V_{k+1}^\pi \qquad V_K^\pi = r_K^\pi$$

where $V_k^\pi = (V_k^\pi(1),\cdots,V_k^\pi(N))$

FACT 2.6 Under a stationary, Markov policy π,

$$P^\pi(X_{k+1},X_{k+2},\cdots|X_k = i) = P^\pi(X_1,X_2,\cdots|X_0 = i), \quad \forall i.$$

2.3 Stochastic Linear Systems

stochastic linear system.

$$x_{k+1} = Ax_k + Bu_k + w_k, \qquad X = \Re^n$$
$$y_k = Cx_k + v_k, \qquad U = \Re^m$$
$$Y = \Re^p$$

A, B and C are matrices of appropriate dimensions,

at time k, $x_k \in X$ denotes state $\quad w_k$ is plant noise
$u_k \in U$ control action $\quad v_k$ is observation
$y_k \in Y$ observation $\quad$ or sensor noise.

`with` probability distributions for $x_0, w_0, w_1, \cdots, v_0, v_1, \cdots,$

ASSUMPTION 2.7 $x_0, w_0, w_1, \cdots, v_0, v_1, \cdots$ are independent random variables.

ASSUMPTION 2.8 x_0 has Gaussian distribution $\mathcal{N}(\bar{x}_0, \Sigma_0)$. w_k has Gaussian distribution $\mathcal{N}(0, Q)$. v_k has Gaussian distribution $\mathcal{N}(0, R)$.

PROPOSITION 2.9 *Suppose A is stable. Then, $\lim_{k\to\infty} \Sigma_k$ exists, is unique and given by a positive semi-definite matrix, Σ_∞ that is a fixed point of the following equation*

$$\Sigma = A\Sigma A^T + Q.$$

3 Finite horizon MDPs

3.1 Introduction

3.2 The Dynamic Programming Algorithm

Algorithm 1 Dynamic Programming

Initial counter: $k = K$.

1. $V_K(x) = r_K(x), \quad \forall x \in X$
2. $V_k(x) = \sup_{u \in U}\{r_k(x,u) + \mathbb{E}[V_{k+1}(f_k(x,u,w_k))]\}, \quad \forall x \in X$
3. $\pi_k(x)$ is maximizer in the above step.
4. while $k > 0$, $k \leftarrow k-1$; Goto Step 2.

Output: $V_0(\cdot)$ and $\pi = (\pi_0, \pi_1, \pi_{K-1})$.

LEMMA 3.2 *In the backward recursive algorithm* (3.4),

$$V_K^\pi(x) = r_K(x), \quad \forall x \in X, \tag{3.4}$$

$$V_k^\pi((x) = r_k(x,u) + \mathbb{E}[V_{k+1}^\pi((f_k(x,u,w_k))], \quad \text{for } k = K-1, K-2, \cdots, 0.$$

$$V_k^\pi(x_k^\pi) = J_k^\pi(x_k^\pi) \quad a.s. \tag{3.5}$$

THEOREM 3.3 (Optimality of Dynamic Programming) *A Markov policy π is optimal if and only if it is a supremizer in Algorithm 1.*

Bellman's Principle of Optimality

```
Optimal Objective Value from k
=
Optimal value (stage reward at k + Optimal Objective Value from k+1)
```

3.3 The Linear Programming approach to DP

$$\min_{\{v_k\}} \sum_{i=1}^{N} v_0(i)$$

$$\text{s.t. } v_k(i) \geq r_k(i,u) + \sum_{j=1}^{N} P_{ij}(u) v_{k+1}(j), \quad \forall i, \forall u, \quad k = 0, \cdots, K-1,$$

$$v_K(i) \geq r_K(i), \quad \forall i.$$

$$v_k(i) = \sup_u\{r_k(i,u) + \sum_{j=1}^{N} P_{ij}(u) v_{k+1}(j)\} \qquad V_K(i) = r_K(i)$$

3.4 Partial Observations and the Belief State

information available at time k.

$$z_k = (y_k, u_{k-1})$$

$$z^k = (z_0, \cdots, z_k)$$

belief of the current state

$$p_{k|k}(x_k|z^k) = P(x_k|z^k)$$

and $p_{k+1|k}(x_{k+1}|z^k, u_k) = P(x_{k+1}|z^k, u_k)$

$$p_{k+1|k+1}(x_{k+1}|z^{k+1}) = \frac{P(y_{k+1}|x_{k+1})p_{k+1|k}(x_{k+1}|z^k, u_k)}{\sum_{x_{k+1}} P(y_{k+1}|x_{k+1})p_{k+1|k}(x_{k+1}|z^k, u_k)}$$

$$= \frac{P(y_{k+1}|x_{k+1})\sum_{x_k} P(x_{k+1}|x_k, u_k)p_{k|k}(x_k|z^k)}{\sum_{x_{k+1}} P(y_{k+1}|x_{k+1})\sum_{x_k} P(x_{k+1}|x_k, u_k)p_{k|k}(x_k|z^k)}$$

$$= \Gamma_k[p_{k|k}(\cdot|z^k), u_k, y_{k+1})]$$

belief (or information) state.

$$\xi_k(z^k) = (p_{k|k}(x_k = i|z^k))_{i=1}^N$$

$$\xi_{k+1}(z^{k+1}) = \Gamma_k[\xi_k(z^k), u_k, y_{k+1}]$$

note

$\xi_k(z^k) \in \Delta(X)$, the space of probability distributions over X

3.5 The DP Algorithm for Partially Observed MDPs

Algorithm 2 Dynamic Programming for POMDPs

Initial counter: $k = K$.

1. $V_K(\xi) = \mathbb{E}[r_K(x)|\xi_K = \xi], \quad \forall \xi \in \Delta(X)$
2. $V_k(\xi) = \sup_{u \in U} \mathbb{E}[r_k(x_k, u) + V_{k+1}(\Gamma_k(\xi_k, y_{k+1}, u))|\xi_k = \xi], \quad \forall \xi \in \Delta(X)$
3. $\pi_k(\xi)$ is a maximizer in the above step.
4. while $k > 0$, $k \leftarrow k - 1$; Goto Step 2.

Output: $V_0(\cdot)$ and $\pi = (\pi_0, \pi_1, \cdots, \pi_{K-1})$.

THEOREM 3.4 *Let π^* be a separated policy such that $\pi_k^*(\xi)$ achieves supremum in Step 2 of Algorithm 2. Then, π^* is an optimal policy and $V_k(\xi_k(z^{*,k})) = J_k^*$ a.s.. Conversely, if π^* is a separated optimal policy, then it is a supermizer in Step 2 of Algorithm 2.*

4 Infinite horizon Discounted MDPs

4.1 Introduction

Let us define

$$V^\pi(x_0) = \mathbb{E}_\pi[\sum_{k=0}^{\infty}\gamma^k r(x_k, u_k)|x_0],$$

then,

$$V^\pi(x_0) = r(x_0, u_0) + \gamma\mathbb{E}[V^\pi(x_1)|x_0],$$

4.2 The Bellman Equation of Optimality

operator $T^\pi : \mathcal{F} \to \mathcal{F}$ For a stationary Markov policy π

$$[T^\pi v](x) := r(x, \pi(x)) + \gamma\mathbb{E}[v(x_1)|x_0 = x]$$

where $\mathcal{F}$ is the space of all non-negative functions from X to $\Re$

Then,

$$T^\pi V^\pi = V^\pi$$

i.e., V^π is a fixed point of the linear operator T^π.

4.3 An Operator Theoretic View

Bellman operator. $T : \mathcal{F} \to \mathcal{F}$ such that

$$[Tv](x) := \sup_u\{r(x, u) + \gamma\mathbb{E}[v(x_1)|x_0 = x]\},$$

. $\mathcal{F}$ is the space of all functions $f : X \to \Re^N$

$$TV_\infty = V_\infty$$ i.e., the V_∞ is a *fixed point*

contraction.

$$T : \mathcal{F} \to \mathcal{F} \qquad \text{for any } v_1, v_2 \in \mathcal{F}.$$

$$||Tv_1 - Tv_2|| \le \gamma || \sum_j (v_1(j) - v_2(j)) P_{ij}(\tilde{u}) || \le \gamma || \max_j |v_1(j) - v_2(j)| || = \gamma ||v_1 - v_2||$$

$$||Tv_1 - Tv_2|| \le \gamma ||v_1 - v_2||$$

$\sum_j P_{ij}(\tilde{u}) = 1$

THEOREM 4.1 (Banach Fixed Point Theorem) *Let $\mathcal{F}$ be a complete normed (Banach) space with norm $||\cdot||$. Let $T : \mathcal{F} \to \mathcal{F}$ be a contraction, i.e.,*

$$||Tv_1 - Tv_2|| \le \gamma ||v_1 - v_2||, \quad \forall v_1, v_2 \in \mathcal{F}, \quad 0 < \gamma < 1.$$

Then, (i) there exists a unique fixed point of T, v^* such that $Tv^* = v^*$, and (ii) for any $v_0 \in \mathcal{F}$, the sequence $(v_0, Tv_0, \cdots, T^k v_0, \cdots)$ has the limit v^*, i.e.,

$$\lim_{n \to \infty} ||T^n v_0 - v^*|| = 0.$$

4.4 Dynamic Programming Algorithms

sup-norm,

$$||v||_\infty = \max_x |v(x)|.$$

Banach space.

$$(\mathcal{F}, ||\cdot||_\infty)$$

with $0 \le r(x, u) \le \bar{R}$. V^π bounded by $\bar{V} := \bar{R}/(1-\gamma)$

$$\mathcal{F} = [0, \bar{V}]^N.$$

$N :=$ cardinality of X

The Value Iteration Algorithm

start with any V_0.

Apply the Bellman operator T to it iteratively, i.e.,

$$V_{k+1}(x) = \sup_u \{ r(x, u) + \gamma \sum_{x'} P_{xx'}(u) V_k(x') \}, \quad \forall x.$$

THEOREM 4.2 *In the Value Iteration algorithm (4.2), (i)*

$$V_n(x) = \sup_{\pi_n} \mathbb{E}^{\pi}[\sum_{k=0}^{n-1} \gamma^k r(x_k, u_k)|x_0 = x],$$

and (ii)

$$V^*(x) = \sup_{\pi_n} \mathbb{E}^{\pi}[\sum_{k=0}^{\infty} \gamma^k r(x_k, u_k)|x_0 = x].$$

Q-value function,

$$Q^*(x,u) = r(x,u) + \gamma \mathbb{E}[V^*(x')|x,u],$$

V^* is the optimal value function

operator G,

$$[GQ](x,u) := r(x,u) + \gamma \mathbb{E}[\sup_{u'} Q(x',u')|x,u].$$

note that $V^*(x) = \sup_u Q^*(x,u), \quad \forall x,$ $Q^* = GQ^*,$

The Q-Value Iteration Algorithm

Start with any Q_0.

Iteratively compute the Q-values,

$$Q_{k+1}(x,u) = r(x,u) + \gamma \sum_{x'} P_{x,x'}(u) \sup_{u'} Q_k(x',u'), \quad \forall x, u.$$

$$\pi^*(x) \in \arg\sup_u Q^*(x,u).$$

The Policy Iteration Algorithm

Start with any (deterministic) policy π_0

policy evaluation

$$\textit{solve} \quad T_{\pi_n} V^{\pi_n} = V^{\pi_n}$$

policy improvement

$$\pi_{n+1}(x) \in \arg\sup_u \{ r(x,u) + \gamma \sum_{x'} P_{xx'}(\pi_n(x)) V^{\pi_n}(x') \} \quad \forall x,$$

repeat until $\pi_{n+1} = \pi_n$,

THEOREM 4.3 *The policy iteration algorithm converges to an optimal policy in a finite number of steps. Moreover.*

$$V^{\pi_{n+1}} > V^{\pi_n},$$

at each stage n, with inequality for all states x and strict inequality for at least one x.

Continuous State Spaces *function approximation*

Choose a set of basis functions $\phi_1(x), \cdots, \phi_m(x)$

Now approximate V^* as $V^*(x) \approx \sum_{j=1}^m w_j \phi_j(x)$

Pick m points $x_1, \cdots, x_m$

Given V_k, evaluate $V_{k+1}(x_1), \cdots, V_{k+1}(x_m)$ as in the *value iteration* algorithm

Now, find weights $w = (w_1, \cdots, w_m)$ that say minimize squared-error, i.e.,

$$\min_w \sum_{i=1}^m (V_{k+1}(x_i) - \sum_{j=1}^m w_j \phi_j(x_i))^2$$

Suppose the solution is $\tilde{w}$. Then,

$$V_{k+1}(x) \approx \sum_{j=1}^m \tilde{w}_j \phi_j(x)$$

4.6 Empirical Dynamic Programming

Empirical Value Iteration (EVI)

$$\hat{V}_{k+1}(x) = \sup_u \{r(x,u) + \gamma \hat{\mathbb{E}}_n[\hat{V}_k(x')|x,u]\}, \quad \forall x, u.$$

$$\textit{with} \quad \hat{\mathbb{E}}_n[V_k(x')|x,u] = \frac{1}{n}\sum_{i=1}^{n} V_k(x_i'),$$

$x_i', i = 1, \cdots, n$ are samples of the next state from state x with action u.

random operator $\hat{T}_n$,

$$\hat{V}_{k+1} = \hat{T}_n \hat{V}_k,$$

$$(\hat{T}_n v)(x) := \sup_u \{r(x,u) + \gamma \hat{\mathbb{E}}_n[v(x')|x,u]\}.$$

By Weak (or Strong) Law of Large numbers, we can expect that $\hat{\mathbb{E}}_n[v(x')|x,u] \to \mathbb{E}[v(x')|x,u]$ in probability (or almost surely) as $n \to \infty$. Indeed, we can show that

$$\lim_{n\to\infty} \mathbb{P}\left(|\hat{T}_n v - Tv| > \epsilon\right) = 0, \quad \text{for any } v.$$

4.7 The Linear Programming Approach

linear program.

$$\begin{array}{ll} \min & \sum_{i=1}^{N} v(i), \\ \text{s.t.} & v(i) \geq r(i,u) + \gamma \sum_{j=1}^{N} P_{ij}(u) v(j), \quad \forall i, u. \end{array}$$

Note that for each $v(i), i = 1, \cdots, N$,

$$v(i) = \sup_u \{r(i,u) + \gamma \sum_{j=1}^{N} P_{ij}(u) v(j)\},$$

4.8 MDPs with Constraints

, constrained MDP problem

$$\max_{\pi} \quad \mathbb{E}[\textstyle\sum_{k=0}^{\infty} \gamma^k r(x_k, u_k)],$$
$$\text{s.t. } \mathbb{E}[\textstyle\sum_{k=0}^{\infty} \gamma^k c(x_k, u_k)] \leq \bar{C}.$$

"occupation measure" $\mu(x, u)$

$$\mu(x, u) = \sum_{k=0}^{\infty} \gamma^k P(x_k = x, u_k = u).$$

discounted probability of occupancy in the State-Action space.

CMDP LP

$$\begin{aligned} \max_{\mu} \quad & \textstyle\sum_{x,u} r(x, u)\mu(x, u), \\ \text{s.t.} \quad & \textstyle\sum_{x,u} c(x, u)\mu(x, u) \leq \bar{C}, \\ & \textstyle\sum_u \mu(x, u) = \nu(x) + \gamma \sum_{x'} \sum_{u'} P_{x'x}(u')\mu(x', u'), \quad \forall x, \\ & \mu(x, u) \geq 0, \ \forall x, u. \end{aligned}$$

5 Infinite horizon Averaged MDPs

5.1 Introduction

infinite horizon average reward

$$J^{\pi} := \liminf_{K\to\infty} \frac{1}{K}\mathbb{E}_{\pi}[\sum_{k=0}^{K-1} r(x_k, u_k)]$$

5.2 The Bellman Equation of Optimality

Let us define over a finite horizon K,

$$V_k^{\pi}(x) := \mathbb{E}_{\pi}[\sum_{l=0}^{k} r(x_l, u_l)|x_0 = x], \quad k = 0, \cdots, K$$

Then, for $u = \pi(x)$,

$$V_K^{\pi}(x) = \{r(x, u) + \mathbb{E}_{\pi}[V_{K-1}^{\pi}(x')|x, u]\}.$$

$$v^{\pi}(x) = \lim_{K\to\infty}[V_K^{\pi}(x) - KJ^{\pi}], \quad \forall x,$$

$$\lim_{K\to\infty} \frac{V_K^{\pi}(x)}{K} = J^{\pi},$$

Bellman's equation of optimality for the average reward case:

$$J^* + v^*(x) = \sup_u\{r(x, u) + \mathbb{E}[v^*(x')|x, u]\}, \tag{5.2}$$

v^* is the optimal relative value function

J^* optimal expected average reward

Note that for large horizon K,

$$V_K^*(x) = KJ^* + v^*(x) + o(1).$$

ASSUMPTION 5.1 The MDP is *unichain*, i.e., for any stationary and Markov policy π, the Markov chain P_π induced by the MDP is irreducible.

Note that an irreducible Markov chain has a single communicating class.

THEOREM 5.2 *Under Assumption* (5.1), *(i) there exists a solution to the Bellman equation* (5.2) *where J^* is unique and w^* is unique upto an additive constant. (ii) Let π^* denote the policy corresponding to the maximizer in* (5.2). *Then, π^* is an optimal policy and J^* is the optimal expected average reward. (iii) If π is an optimal policy and $J^* = J^\pi$, then it satisfies* (5.2).

LEMMA 5.3 *Suppose there exist J^* and v^* that satisfy the average Bellman equation* (5.2). *Then, J^* is the optimal expected average reward and the supremizing policy π^* is optimal.*

LEMMA 5.4 *Under Assumption* (5.1), *there exist solutions (J^*, v^*) to* (5.2).

LEMMA 5.5 *Suppose π^* is an optimal policy. Then, it satisfies the average Bellman equation* (5.2).

Q-relative value function, $q(x,u)$.

$$J^* + q^*(x,u) = r(x,u) + \sum_{x'} P_{xx'}(u) \sup_{u'} q^*(x',u').$$

5.4 DP Algorithms for Average MDPs

Relative Value Iteration Algorithm

Algorithm 3 Relative Value Iteration

Input: $V_0(x) = 0$, $v_0(x) = 0$ and a reference state x_{ref}.

1. $V_{k+1}(x) = \sup_{u \in U}\{r(x,u) + \mathbb{E}[v_k(f_k(x,u,w_k))]\}, \quad \forall x \in X$
2. $\pi_{k+1}(x)$ is maximizer in the above step.
3. $v_{k+1}(x) = V_{k+1}(x) - V_{k+1}(x_{ref})$
4. $k \leftarrow k+1$; Goto Step 2.

Output: $v^*(\cdot)$ and π^*.

relative Q-value iteration (RQVI)

Algorithm 3 Relative Value Iteration

Input: $V_0(x) = 0$, $v_0(x) = 0$ and a reference state x_{ref}.

1. $Q_{k+1}(x,u) = r(x,u) + \mathbb{E}[\sup_{u' \in U} q_k(f_k(x,u,w_k),u')], \quad \forall x \in X$.
2. $\pi_{k+1}(x)$ is maximizer in the above step.
3. $q_{k+1}(x,u) = Q_{k+1}(x,u) - Q_{k+1}(x_{ref}, u_{ref}), \quad \forall x, u$.
4. $k \leftarrow k+1$; Goto Step 2.

Output: $q^*(\cdot)$ and π^*.

The Policy Iteration Algorithm

Start with any (deterministic) policy π_0.

policy evaluation obtain v^{π_k}

$$J^{\pi_k} + v^{\pi_k}(x) = \{r(x,\pi_k(x)) + \mathbb{E}[v^{\pi_k}(x')|x,\pi_k(x)]\}.$$

setting $v^{\pi_k}(x_{ref}) = 0$ for a reference state x_{ref}

policy improvement

$$\pi_{k+1}(x) \in \arg\sup_u \{r(x,u) + \sum_{x'} P_{xx'}(\pi_n(x)) v^{\pi_k}(x')\} \quad \forall x.$$

repeat until $\pi_{k+1} = \pi_k$,

5.5 Linear Program for Average MDPs

linear program.

$$\min_{J^*, w} \quad J^*$$

$$\text{s.t.} \quad J^* + v(i) \geq r(i,u) + \textstyle\sum_{j=1}^N P_{ij}(u) v(j), \quad \forall i, u.$$

5.6 Blackwell's Optimality criterion: Average reward as a limit of Discounted reward

THEOREM 5.6 *Let V^γ be the optimal value function for the discounted reward MDP with discount factor $\gamma \in (0,1)$. Then, if*

$$|V^\gamma(x') - V^\gamma(x)| < M, \forall x, x',$$

for some $M < \infty$, then there exists a sequence $\{\gamma_l\}$, $0\gamma_l < 1$ and $\gamma_l \to 1$ as $l \to \infty$ such that limits J^* and v^* exist and satisfy the average reward Bellman equation (5.2).

Bolzano-Weirstrass Theorem,

there exists a sequence $\{\gamma_l\}$,

and the limits

$$J^* =: \lim_{l \to \infty} (1-\gamma_l) V^{\gamma_l}(1) \quad \text{and}$$

$$v^*(x) =: \lim_{l \to \infty} [V^{\gamma}(x) - V^{\gamma}(1)]$$

exist.

5.7 MDPs with Constraints

constrained MDP problem of the following kind:

$$\max_{\pi} \quad \lim_{K\to\infty} \tfrac{1}{K}\mathbb{E}[\textstyle\sum_{k=0}^{K-1} r(x_k, u_k)],$$

$$\text{s.t.} \lim_{K\to\infty} \tfrac{1}{K}\mathbb{E}[\textstyle\sum_{k=0}^{K-1} c(x_k, u_k)] \le \bar{C}.$$

"occupation measure" $\mu(x,u)$,

$$\mu(x,u) := \lim_{K\to\infty} \frac{1}{K} \sum_{k=0}^{K-1} P(x_k = x, u_k = u).$$

denoting the initial state distribution by ν,

CMDP LP average reward

$$\max_{\mu} \sum_{x,u} r(x,u)\mu(x,u),$$

$$\text{s.t.} \sum_{x,u} c(x,u)\mu(x,u) \le \bar{C},$$

$$\sum_{u} \mu(x,u) = \sum_{u} \sum_{x',u'} \sum_{u'} P_{x'x}(u')\mu(x',u'), \quad \forall x,$$

$$\mu(x,u) \ge 0, \ \forall x, u.$$

5.8 An EDP Algorithm

Empirical Ralative Value Iteration (ERVI)

$$\hat{V}_{k+1}(x) = \sup_u \{r(x,u) + \hat{\mathbb{E}}_n[\hat{v}_k(x')|x,u]\}, \quad \forall x, u. \hat{v}_{k+1}(x) \quad \hat{V}_{k+1}(x) - \hat{V}_{k+1}(x_{ref}),$$

x_{ref} is a reference state (chosen arbitrarily)

with $$\hat{\mathbb{E}}_n[v_k(x')|x,u] = \frac{1}{n}\sum_{i=1}^{n} v_k(x_i')$$

$x_i', i = 1, \cdots, n$ are samples of the next state from state x with action u.

random operator $\hat{T}_n$

$$\hat{v}_{k+1} = \hat{T}_n \hat{v}_k,$$

$$(\hat{T}_n v)(x) := \sup_u \{r(x,u) + \gamma\hat{\mathbb{E}}_n[v(x')|x,u] - \sup_u \{r(x_{ref},u) + \gamma\hat{\mathbb{E}}_n[v(x')|x_{ref},u]\}.$$