Recitation 3: Homework 1

Yafei Hu and Justin Kiefel

Problem 1: Value Iteration & Policy Iteration

1.1: Contraction Mapping

An operator F on a normed vector space $\mathcal{X}$ is a γ-contraction, for $0 < \gamma < 1$ provided for all $x, y \in \mathcal{X}$:

$$||F(x) - F(y)|| \leq \gamma ||x - y||$$

Theorem (Contraction mapping)
For a γ-contraction F in a complete normed vector space $\mathcal{X}$:

- F converges to a unique fixed point in $\mathcal{X}$,
- at a linear convergence rate γ.

Problem 1: Value Iteration & Policy Iteration

Policy Evaluation

Iterative Policy Evaluation, for estimating $V \approx v_\pi$

Input π, the policy to be evaluated
Algorithm parameter: a small threshold $\theta > 0$ determining accuracy of estimation
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(terminal) = 0$

Loop:
  $\Delta \leftarrow 0$
  Loop for each $s \in \mathcal{S}$:
    $v \leftarrow V(s)$
    $V(s) \leftarrow \sum_a \pi(a|s) \sum_{s',r} p(s', r \,|\, s, a)\big[r + \gamma V(s')\big]$
    $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
until $\Delta < \theta$

Problem 1: Value Iteration & Policy Iteration

Policy Iteration

Note the difference between sync and sync PI

Policy Iteration (using iterative policy evaluation) for estimating $\pi \approx \pi_*$

1. Initialization
   $V(s) \in \mathbb{R}$ and $\pi(s) \in \mathcal{A}(s)$ arbitrarily for all $s \in \mathcal{S}$

2. Policy Evaluation
   Loop:
   $\quad \Delta \leftarrow 0$
   $\quad$ Loop for each $s \in \mathcal{S}$:
   $\quad\quad v \leftarrow V(s)$
   $\quad\quad V(s) \leftarrow \sum_{s',r} p(s', r|s, \pi(s))\big[r + \gamma V(s')\big]$
   $\quad\quad \Delta \leftarrow \max(\Delta, |v - V(s)|)$
   until $\Delta < \theta$ (a small positive number determining the accuracy of estimation)

3. Policy Improvement
   policy-stable $\leftarrow$ *true*
   For each $s \in \mathcal{S}$:
   $\quad$ *old-action* $\leftarrow \pi(s)$
   $\quad \pi(s) \leftarrow \arg\max_a \sum_{s',r} p(s', r|s, a)\big[r + \gamma V(s')\big]$
   $\quad$ If *old-action* $\neq \pi(s)$, then *policy-stable* $\leftarrow$ *false*
   If *policy-stable*, then stop and return $V \approx v_*$ and $\pi \approx \pi_*$; else go to 2

Problem 1: Value Iteration & Policy Iteration

Value Iteration

Note the difference between sync and sync PI

Value Iteration, for estimating $\pi \approx \pi_*$

Algorithm parameter: a small threshold $\theta > 0$ determining accuracy of estimation
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(terminal) = 0$

Loop:
| $\Delta \leftarrow 0$
| Loop for each $s \in \mathcal{S}$:
| $v \leftarrow V(s)$
| $V(s) \leftarrow \max_a \sum_{s',r} p(s', r \mid s, a)\big[r + \gamma V(s')\big]$
| $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
until $\Delta < \theta$

Output a deterministic policy, $\pi \approx \pi_*$, such that
$\pi(s) = \operatorname{argmax}_a \sum_{s',r} p(s', r \mid s, a)\big[r + \gamma V(s')\big]$

Problem 1: Value Iteration & Policy Iteration

Synchronous and Asynchronous Policy Iteration/Value Iteration

- Synchronous value iteration stores two copies of value function
  - for all s in $\mathcal{S}$

$$v_{new}(s) \leftarrow \max_{a \in \mathcal{A}} \left(r(s,a) + \gamma \sum_{s' \in \mathcal{D}} p\left(s' \mid s, a\right) v_{old}\left(s'\right) \right)$$

$$v_{old} \leftarrow v_{new}$$

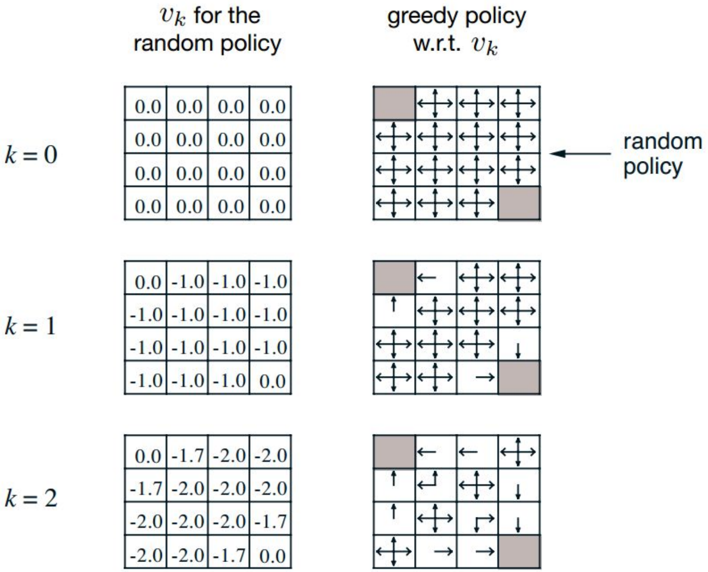
- In-place value iteration only stores one copy of value function
  - for all s in $\mathcal{S}$

$$v(s) \leftarrow \max_{a \in \mathcal{A}} \left(r(s,a) + \gamma \sum_{s' \in \mathcal{S}} T\left(s' \mid s, a\right) v\left(s'\right) \right)$$

Problem 1: Value Iteration & Policy Iteration

Synchronous and Asynchronous Policy Iteration/Value Iteration

A tabular state value function showing the difference, like



Problem 1: Value Iteration & Policy Iteration

Problem 1.5: Manhattan distance as heuristic function

```
function heuristic(node) =
    dx = abs(node.x - goal.x)
    dy = abs(node.y - goal.y)
    return D * (dx + dy)
```

Ref: http://theory.stanford.edu/~amitp/GameProgramming/Heuristics.html

Problem 2: Bandits

Estimating Expected Reward

$$\mathbb{E}\{R_t\} = \frac{1}{20}\Sigma_{k=1}^{20} R_t^k$$

- Average of rewards received at a given time step
- Unbiased
- High Variance

$$\mathbb{E}\{R_t\} = \frac{1}{20}\Sigma_{k=1}^{20} \mathbb{E}\{r^k(A_t^k)|\pi_t^k\}$$

- Average of expected rewards conditioned on the policy
- Unbiased
- Lower Variance
- Remember to still use R_t for the agent's update

Efficient Q-Updates

$$
\begin{aligned}
Q_{n+1} &= \frac{1}{n}\sum_{i=1}^{n} R_i \longleftarrow \text{Don't Use This} \\
&= \frac{1}{n}\left(R_n + \sum_{i=1}^{n-1} R_i\right) \\
&= \frac{1}{n}\left(R_n + (n-1)\frac{1}{n-1}\sum_{i=1}^{n-1} R_i\right) \\
&= \frac{1}{n}\Big(R_n + (n-1)Q_n\Big) \\
&= \frac{1}{n}\Big(R_n + nQ_n - Q_n\Big) \\
\text{Use This} \longrightarrow &= Q_n + \frac{1}{n}\Big[R_n - Q_n\Big],
\end{aligned}
$$

Problem 2.7 - Correlated Rewards

I.I.D. Rewards

$$r(k) \sim \mathcal{N}(\mu, \sigma^2)\ \forall k \in [K]$$

Correlated Rewards

$$[r(1) \ldots r(K)]^T \sim \mathcal{N}(\mu_0, \Sigma_0)$$

Non-Diagonal