Reinforcement Learning
Reminders

• Homework 7: HMMs
  – Out: Wed, Apr 04
  – Due: Mon, Apr 16 at 11:59pm

• Schedule Changes
  – Lecture on Fri, Apr 13
  – Recitation on Mon, Apr 23
Learning Paradigms

Whiteboard

– Supervised
  • Regression
  • Classification
  • Binary Classification
  • Structured Prediction
– Unsupervised
– Semi-supervised
– Online
– Active Learning
– Reinforcement Learning
REINFORCEMENT LEARNING
Examples of Reinforcement Learning

• How should a robot behave so as to optimize its “performance”? (Robotics)

• How to automate the motion of a helicopter? (Control Theory)

• How to make a good chess-playing program? (Artificial Intelligence)
Autonomous Helicopter

Video:

https://www.youtube.com/watch?v=VCdxqnofcnE
Robot in a room

- reward +1 at [4,3], -1 at [4,2]
- reward -0.04 for each step
- what’s the strategy to achieve max reward?
- what if the actions were NOT deterministic?
History of Reinforcement Learning

• Roots in the psychology of animal learning (Thorndike, 1911).

• Another independent thread was the problem of optimal control, and its solution using dynamic programming (Bellman, 1957).

• Idea of temporal difference learning (on-line method), e.g., playing board games (Samuel, 1959).

• A major breakthrough was the discovery of Q-learning (Watkins, 1989).
What is special about RL?

• RL is learning how to map states to actions, so as to maximize a numerical reward over time.

• Unlike other forms of learning, it is a multistage decision-making process (often Markovian).

• An RL agent must learn by trial-and-error. (Not entirely supervised, but interactive)

• Actions may affect not only the immediate reward but also subsequent rewards (Delayed effect).
Elements of RL

• A policy
  - A map from state space to action space.
  - May be stochastic.
• A reward function
  - It maps each state (or, state-action pair) to a real number, called reward.
• A value function
  - Value of a state (or, state-action pair) is the total expected reward, starting from that state (or, state-action pair).
Policy

![Diagram showing a grid with arrows and numbers (+1 and -1)]
Reward for each step -2
Reward for each step: -0.1
The Precise Goal

• To find a policy that maximizes the Value function. – transitions and rewards usually not available

• There are different approaches to achieve this goal in various situations.

• Value iteration and Policy iteration are two more classic approaches to this problem. But essentially both are dynamic programming.

• Q-learning is a more recent approaches to this problem. Essentially it is a temporal-difference method.
MARKOV DECISION PROCESSES
Markov Decision Process

Whiteboard

- Components: states, actions, state transition probabilities, reward function
- Markovian assumption
- MDP Model
- MDP Goal: Infinite-horizon Discounted Reward
- deterministic vs. nondeterministic MDP
- deterministic vs. stochastic policy
Exploration vs. Exploitation

Whiteboard

– Explore vs. Exploit Tradeoff
– Ex: k-Armed Bandits
– Ex: Traversing a Maze
FIXED POINT ITERATION
Fixed Point Iteration for Optimization

• Fixed point iteration is a general tool for solving systems of equations
• It can also be applied to optimization.

<table>
<thead>
<tr>
<th>$J(\theta)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\frac{dJ(\theta)}{d\theta_i} = 0 = f(\theta)$</td>
</tr>
<tr>
<td>$0 = f(\theta) \Rightarrow \theta_i = g(\theta)$</td>
</tr>
<tr>
<td>$\theta_i^{(t+1)} = g(\theta^{(t)})$</td>
</tr>
</tbody>
</table>

1. Given objective function:
2. Compute derivative, set to zero (call this function $f'$).
3. Rearrange the equation s.t. one of parameters appears on the LHS.
4. Initialize the parameters.
5. For $i$ in $\{1,\ldots,K\}$, update each parameter and increment $t$:
6. Repeat #5 until convergence
Fixed Point Iteration for Optimization

- Fixed point iteration is a general tool for solving systems of equations.
- It can also be applied to optimization.

<table>
<thead>
<tr>
<th>$J(x) = \frac{x^3}{3} + \frac{3}{2}x^2 + 2x$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\frac{dJ(x)}{dx} = f(x) = x^2 - 3x + 2 = 0$</td>
</tr>
<tr>
<td>$\Rightarrow x = \frac{x^2 + 2}{3} = g(x)$</td>
</tr>
<tr>
<td>$x \leftarrow \frac{x^2 + 2}{3}$</td>
</tr>
</tbody>
</table>

1. Given objective function:
2. Compute derivative, set to zero (call this function $f$).
3. Rearrange the equation s.t. one of parameters appears on the LHS.
4. Initialize the parameters.
5. For $i$ in $\{1, \ldots, K\}$, update each parameter and increment $t$:
6. Repeat #5 until convergence.
Fixed Point Iteration for Optimization

We can implement our example in a few lines of python.

\[
J(x) = \frac{x^3}{3} + \frac{3}{2}x^2 + 2x \\
\frac{dJ(x)}{dx} = f(x) = x^2 - 3x + 2 = 0 \\
\Rightarrow x = \frac{x^2 + 2}{3} = g(x) \\
x \leftarrow \frac{x^2 + 2}{3}
\]
Fixed Point Iteration for Optimization

$\text{python fixed-point-iteration.py}$

\begin{align*}
\text{i} &= 0 \quad x=0.0000 \quad f(x)=2.0000 \\
\text{i} &= 1 \quad x=0.6667 \quad f(x)=0.4444 \\
\text{i} &= 2 \quad x=0.8148 \quad f(x)=0.2195 \\
\text{i} &= 3 \quad x=0.8880 \quad f(x)=0.1246 \\
\text{i} &= 4 \quad x=0.9295 \quad f(x)=0.0755 \\
\text{i} &= 5 \quad x=0.9547 \quad f(x)=0.0474 \\
\text{i} &= 6 \quad x=0.9705 \quad f(x)=0.0304 \\
\text{i} &= 7 \quad x=0.9806 \quad f(x)=0.0198 \\
\text{i} &= 8 \quad x=0.9872 \quad f(x)=0.0130 \\
\text{i} &= 9 \quad x=0.9915 \quad f(x)=0.0086 \\
\text{i} &= 10 \quad x=0.9944 \quad f(x)=0.0057 \\
\text{i} &= 11 \quad x=0.9963 \quad f(x)=0.0038 \\
\text{i} &= 12 \quad x=0.9975 \quad f(x)=0.0025 \\
\text{i} &= 13 \quad x=0.9983 \quad f(x)=0.0017 \\
\text{i} &= 14 \quad x=0.9989 \quad f(x)=0.0011 \\
\text{i} &= 15 \quad x=0.9993 \quad f(x)=0.0007 \\
\text{i} &= 16 \quad x=0.9995 \quad f(x)=0.0005 \\
\text{i} &= 17 \quad x=0.9997 \quad f(x)=0.0003 \\
\text{i} &= 18 \quad x=0.9998 \quad f(x)=0.0002 \\
\text{i} &= 19 \quad x=0.9999 \quad f(x)=0.0001 \\
\text{i} &= 20 \quad x=0.9999 \quad f(x)=0.0001
\end{align*}

\[J(x) = \frac{x^3}{3} + \frac{3}{2}x^2 + 2x\]

\[\frac{dJ(x)}{dx} = f(x) = x^2 - 3x + 2 = 0\]

\[\Rightarrow x = \frac{x^2 + 2}{3} = g(x)\]

\[x \leftarrow \frac{x^2 + 2}{3}\]
VALUE ITERATION
Definitions for Value Iteration

Whiteboard

- State trajectory
- Value function
- Bellman equations
- Optimal policy
- Optimal value function
- Computing the optimal policy
- Ex: Path Planning