

# Homework 7: Markov Decision Processes

*Due: 5:00 PM, Apr 20, 2009*

## Contents

<b>1</b>	<b>Expected Minimax</b>	<b>1</b>
<b>2</b>	<b>Game Iteration</b>	<b>2</b>
<b>3</b>	<b>Interpreting Discounting</b>	<b>2</b>
<b>4</b>	<b>Gambler's Ruin, Revisited</b>	<b>2</b>
<b>5</b>	<b>Model-Based Learning</b>	<b>3</b>
<b>6</b>	<b>Deterministic <math>Q</math>-Learning</b>	<b>3</b>

## Objectives

By the end of this homework, you will understand:

1. why deterministic  $Q$ -learning converges
2. how discounting relates to dying

By the end of this homework, you will be able to:

1. avert disaster in a casino

## Practice

### 1 Expected Minimax

The goal of this question is to extend the MINIMAX algorithm to games of chance, like backgammon and monopoly.

- (a) Formally define games of chance by extending the definition of game trees.
- (b) Extend the MINIMAX algorithm to take as input a game of chance.

## 2 Game Iteration

The goal of this question is to extend value and policy iteration to games of chance, like backgammon and monopoly.

- (a) Formally define games of chance by extending the definition of MDPs.
- (b) Extend the value iteration algorithm to take as input a game of chance.
- (c) Extend the policy iteration algorithm to take as input a game of chance.

## Problems

### 3 Interpreting Discounting

- (a) Show that for all Markov reward processes, the value function  $V$  is well-defined (*i.e.*, finite-valued) if  $0 \leq \gamma < 1$ .
- (b) Show that there exists a Markov reward process *s.t.* the value function  $V$  is not necessarily well-defined if  $\gamma = 1$ .
- (c) Prove that any Markov reward process  $M_1$ , which for discount factor  $0 \leq \gamma < 1$  yields value function  $V_1$ , can be transformed into another Markov reward process  $M_2$ , *s.t.* the *undiscounted* value function  $V_2 = V_1$ . [Hint: Introduce a zero-reward, absorbing state END in  $M_2$  *s.t.* all state-action pairs in  $M_1$  transition to END with probability  $1 - \gamma$ .]

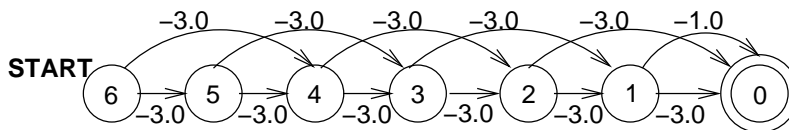
### 4 Gambler's Ruin, Revisited

Consider the following controlled version of *Gambler's Ruin* in which the gambler places bets on the outcome of a biased coin flip. Assume the gambler's worth is between 0 and  $N$  (*i.e.*,  $S = \{0, 1, \dots, N\} \cup \{\text{END}\}$ ). At each state  $s$ , the gambler stakes some amount  $n$  in the range  $A = \{1, \dots, \min\{s, N - s\}\}$ . The coin turns up heads with probability  $p$  and tails with probability  $1 - p$ . If the coin turns up heads, the gambler wins the amount he stakes (*i.e.*, he transitions to state  $s + n$ ); otherwise, the gambler loses the amount he stakes (*i.e.*, he transitions to state  $s - n$ ). A reward of 1 is received upon transitioning to state  $N$ ; all other rewards are 0. The gambler transitions from states 0 and  $N$  to the absorbing state END, deterministically. The constraints on the gambler's range of actions ensure that (i) he stakes at least \$1 but no more than his worth; (ii) he stakes no more than  $N - s$ , preventing his worth from ever exceeding  $N$ .

- (a) Draw the corresponding MDP, assuming the gambler's worth is limited to \$5 (*i.e.*,  $N = 5$ ).
- (b) What is the optimal policy if  $p < 0.5$ ,  $N = 100$ , and  $\gamma = 1$ . Why? (Solve this problem by implementing value iteration or policy iteration.)
- (c) What is the optimal policy if  $p > 0.5$ ,  $N = 100$ , and  $\gamma = 1$ . Why?

## 5 Model-Based Learning

The figure below depicts a Markov reward process with seven states. Every state except the terminal state has two possible successors, each of which occurs with probability 0.5. Rewards are associated with transitions as shown.



Suppose the following sample trajectories are observed:

1.  $6 \xrightarrow{-3} 5 \xrightarrow{-3} 4 \xrightarrow{-3} 3 \xrightarrow{-3} 2 \xrightarrow{-3} 1 \xrightarrow{-1} 0$
2.  $6 \xrightarrow{-3} 4 \xrightarrow{-3} 2 \xrightarrow{-3} 0$

Assume the values  $V(s)$  are initialized to 0, the learning rate  $\alpha = 0.5$ , and the discount factor  $\gamma = 1$ .

- (a) After learning from the first of these trajectories, what value does Monte-Carlo learning assign to each state?
- (b) After learning from the second of these trajectories, what value does TD learning assign to each state?
- (c) An alternative to TD or Monte-Carlo learning is to use the observed sample trajectories to construct a “best guess” at the model that generated those trajectories, and then to solve that model using policy evaluation.

The transition probabilities of the “best guess” model are defined as follows:

$$P(s' | s) = \frac{\text{number of transitions } s \rightarrow s' \text{ in all observed trajectories}}{\text{total number of visits to state } s \text{ in all observed trajectories}}$$

The rewards model  $R(s, s')$  of state transitions out of  $s$  into  $s'$  is given by:

$$R(s, s') = \text{the mean of all rewards observed on transitions } s \rightarrow s'$$

Draw the “best-guess” model corresponding to the two observed trajectories, and solve it using policy evaluation.

## 6 Deterministic Q-Learning

The Q-learning update rule for deterministic MDPs is as follows:

$$Q(s, a) \leftarrow R(s, a) + \gamma \max_{a'} Q(s', a')$$

Prove that Q-learning converges in deterministic MDPs.