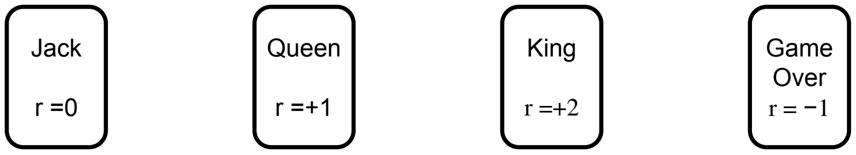
CS 188 Section Handout: Reinforcement Learning

1 Value Iteration

Here's a very simple high-low card game:

1. Draw a card from the top of the deck.
2. Guess whether the next drawn card will be higher or lower.
3. Draw the next card: if the guess was correct, repeat from Step 2; otherwise, game over.

Suppose that we are drawing with replacement, and we use just the face cards. To formulate an MDP for this game, let us suppose that the first card drawn is always a Jack; also, we win \$2 for every correctly guessed King, \$1 for every correctly guessed Queen, \$0 for Jacks, and we pay \$1 when the game is over. Provide the transition model for this MDP:



Passive Learning: Iterate the Bellman updates to estimate the utility of a state:

$$U_{i+1}(s) \leftarrow R(s) + \max_a \sum_{s'} T(s,a,s')U_i(s')$$

i	U_i(Jack)	U_i(Queen)	U_i(King)	U_i(Game Over)
0	0	0	0	0
1	0	1	2	-1
2				-1
3				-1
4				-1

Apply: The optimal policy is found by maximizing the expected utility of subsequent states:

$$\pi^*(s) = \arg\max_a \sum_{s'} T(s,a,s')U(s')$$

π^*(Jack)	π^*(Queen)	π^*(King)

2 Q-learning

Consider this variation of Blackjack:

1. The object is to get as close to 21 points without "busting", and beat the dealer's hand.
2. The available actions are *hit* (draw one more card) or *stay* (play your current hand against the dealer's).
3. The dealer's hand is blind (but he plays by hitting up to 16, then staying).
4. You win \$1 for beating the dealer; lose \$1 for busting or losing; and \$0 for a tie.
5. Your initial hand always totals 16 points.

MDP: Can you specify the MDP representation, including the transition model and reward function?

Initialize: Let's initialize the Q-function arbitrarily:

s	a	$Q(s,a)$
16	hit	
16	stay	
17	hit	
17	stay	
18	hit	
18	stay	
19	hit	
19	stay	
20	hit	
20	stay	

Active Learning: Use the Q-learning algorithm:

```
For each episode:
    $s \leftarrow$ initial state.
    While $s \neq$ terminal state:
        Choose an action $a$, executed in $s$ and resulting in $s'$.
        $Q(s,a) \leftarrow \frac{1}{2}\left[Q(s,a) + R(s) + \max_{a'} Q(s',a')\right]$
        $s \leftarrow s'$
```

Note: This is a fairly simplified version of reinforcement learning in which the learning rate $\alpha = 1/2$, the discount factor $\gamma = 1$, and the exploration function is unspecified.