CS 101.2 - Active Learning
Problem Set 1

Handed out:	14 Jan 2009
Due:	29 Jan 2009

---

1 MDP Value Iteration Algorithm

1. Implement the value iteration algorithm given in the lecture. That is, solve Bellman's equation using dynamic programming. Run your algorithm on the "Becoming Rich and Famous" MDP given in the lecture notes, using discount factor $\gamma = 0.95$. What is the optimal policy and what are the optimal values $V(s)$? Plot the Cauchy sequence

$$\max_s |V_t(s) - V_{t-1}(s)| \tag{1}$$

   as a function of t in order to diagnose convergence.

2. Suppose that instead of optimizing the discounted reward (as discussed in the lecture)

$$\mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r(S_i, \pi(S_i))],$$

   we want to optimize the undiscounted reward over a finite number of T time steps:

$$\mathbb{E}[\sum_{t=0}^{T} r(S_i, \pi(S_i, t))].$$

   In contrast to the undiscounted setting, in the finite horizon setting, the optimal policy can choose a different action for each state, dependent on the current time step. Formally, in the finite horizon setting policies are mappings

$$\pi : S \times \{0, \ldots, T\} \to A,$$

   where $\pi(s, t) = a$ means that the policy π chooses action a if it is in state s at time t. Show how value iteration can be modified to find an optimal policy for the finite horizon setting. *Hint:* As in the discounted setting, use dynamic programming to compute the t-step value function $V_t(s)$ which denotes the maximum expected sum of undiscounted reward that can be accrued within the next t time steps, starting with $t = 0$.

2 An optimal solution for Bandit problems with priors

This problem explores a connection between the bandit problem and Markov Decision Processes, which allows to compute optimal arm pulling strategies for certain bandit problems.

For simplicity we will concentrate on a two-armed problem in which the payoff at each time n is either 1 or 0.

Recall that an MDP is defined by a set of states S, a set of actions A, the reward $r(s,a)$ which is a random variable dependent on the state and the action, and transition probabilities $P(s'|s,a)$. The possible actions are 1 and 2, indicating the arm that is pulled. We will define the state to be $s = (n_1, n_2; h_1, h_2)$ where h_i is the accumulated reward associated with pulling arm i, and n_i is the total number of times arm i has been pulled.

In order to complete the connection, we must specify transition probabilities $P(s'|s,a)$ for the MDP. We will assume that arm i pays 1 with probability μ_i and pays 0 with probability $1-\mu_i$. Given μ_i, the probability that arm i pays total reward h_i after n_i pulls is

$$p(h_i|\mu_i; n_i) = \binom{n_i}{h_i} \mu_i^{h_i} (1-\mu_i)^{n_i - h_i}. \tag{2}$$

However, we do not know what μ_i is. Rather than estimate a confidence interval for this parameter (as in the UCB algorithm), we will instead take the Bayesian approach and maintain a probability distribution over μ_i which encodes our uncertainty about its value. This distribution will be updated based on the rewards paid by arm i. We will place a *prior* distribution over μ_i known as the Beta distribution with parameters α, β:

$$p(\mu_i) = \text{Beta}(\mu_i; \alpha, \beta) = \frac{1}{B(\alpha,\beta)} \mu_i^{\alpha-1} (1-\mu_i)^{\beta-1} \tag{3}$$

where

$$B(\alpha,\beta) = \int_0^1 x^{\alpha-1}(1-x)^{\beta-1} dx \tag{4}$$

is the normalization constant known as the *Beta function*. The Beta distribution is defined over the range $[0,1]$. (If $\alpha = \beta = 1$ then this is equivalent to a uniform distribution.) The prior distribution encodes our belief about the payoff distribution μ_i before we've pulled arm i.

1. We will use Bayes' rule

$$p(\mu_i|h_i; n_i) = \frac{p(h_i|\mu_i; n_i) p(\mu_i)}{\int_0^1 p(h_i|\mu_j; n_i) p(\mu_j) d\mu_j} \tag{5}$$

   to update the distribution of μ_i given that we've pulled arm i a total of n_i times and received pay off h_i. Show that

$$p(\mu_i|h_i; n_i) = \text{Beta}(\alpha + h_i, \beta + n_i - h_i). \tag{6}$$

2. Now we have a distribution over μ_i that takes into account both our prior belief about the payoff distributions and the evidence gained by pulling arms. In order to formulate the MDP, we need the reward distribution for arm i conditioned on only the state. In order to get this we *marginalize* out μ_i.

Show that:

$$p(r_i = 1|h_i, n_i) = \int_0^1 p(r_i = 1|\mu_i)p(\mu_i|h_i; n_i) = \frac{h_i + \alpha}{\alpha + \beta + n_i} \tag{7}$$

(The following facts about the Beta function may be helpful: $B(a, b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$ and $\Gamma(a + 1) = a\Gamma(a)$).

3. Specify the transition probabilities based on the result Eq. (7). Draw a diagram to illustrate the MDP. How many states does this MDP have, if we stop playing after T trials?

4. Explain how the undiscounted value iteration algorithm from Problem 2.2 can be applied to solve for an optimal policy π^* in order to maximize the T-step undiscounted reward

$$\mathbb{E}[\sum_{t=0}^{T} r(S_i, \pi(S_i, t))].$$

5. Sketch how the algorithm can be extended to handle K arms for $K > 2$. What will the complexity of the resulting algorithm (in terms of K and T)?

3 Analysis of the ϵ_n-Greedy Policy

Consider the ϵ_n-Greedy policy for the K-armed bandit problem. The algorithm repeats the following procedure at each time n: Play the machine with highest current average reward $j = \arg\max_j \bar{X}_j$ with probability $(1 - \epsilon_n)$. With probability ϵ_n play a random arm. Use $\epsilon_n = \min\{1, \frac{cK}{d^2 n}\}$ where $c > 0$ and $0 < d < \min_{i:\mu_i<\mu^*} \Delta_i$.

Define $P(I_n = j)$ as the probability that sub-optimal machine j was chosen at time n. We will prove that

$$P(I_n = j) \leq \frac{\epsilon_n}{K} + o(\frac{1}{n}). \tag{8}$$

Hint: You can prove each of the following subproblems independently of the others.

1. Show that:

$$P(I_n = j) \leq \frac{\epsilon_n}{K} + P(\bar{X}_j \geq \bar{X}^*). \tag{9}$$

2. Now we need to bound the second term on the right hand side of eq. 9. We will use tail bounds to do this. First, show that:

$$\begin{aligned} P(\bar{X}_j \geq \bar{X}^*) \leq & P(\bar{X}_j \geq \mu_j + \frac{\Delta_j}{2}) \\ & + P(\bar{X}^* \leq \mu^* - \frac{\Delta_j}{2}) \end{aligned} \tag{10}$$

Hint: Prove the following:

$$P(A \geq B) \leq P(A \geq a) + P(B \leq a) \tag{11}$$

for random variables A and B defined on $[0, \infty)$ and any $a > 0$.

3. We can now handle each of the two terms on the right hand side of eq. 10 using the same analysis. Focusing on the first term, we have:

$$P(\bar{X}_j \geq \mu_j + \frac{\Delta_j}{2}) = \sum_{t=1}^{n} P(T_j = t | \bar{X}_{jt} \geq \mu_j + \frac{\Delta_j}{2}) P(\bar{X}_{jt} \geq \mu_j + \frac{\Delta_j}{2}), \tag{12}$$

where T_j indicates the number of times arm j has been pulled and $\bar{X}_{jt} = \frac{1}{t}\sum_{i=1}^{t} X_{ji}$ is the average reward from machine j after it's been pulled t times. Use the Chernoff-Hoeffding bound to show that

$$P(\bar{X}_j \geq \mu_j + \frac{\Delta_j}{2}) \leq \sum_{t=1}^{n} P(T_j = t | \bar{X}_{jt} \geq \mu_j + \frac{\Delta_j}{2}) e^{\frac{-\Delta_j^2 t}{2}}. \tag{13}$$

The exponential terms in the sum will decay rapidly for sufficiently large values of t, but the first exponential terms are potentially large. Our strategy is then to split the sum into two parts and bound each sum seperately. Define the boundary value $\lfloor x_0 \rfloor$ (we will determine x_0 in the next section). Show that

$$\sum_{t=1}^{n} P(T_j = t | \bar{X}_{jt} \geq \mu_j + \frac{\Delta_j}{2}) e^{\frac{-\Delta_j^2 t}{2}} \leq \sum_{t=1}^{\lfloor x_0 \rfloor} P(T_j = t | \bar{X}_{jt} \geq \mu_j + \frac{\Delta_j}{2}) + \frac{2}{\Delta_j^2} e^{\frac{-\Delta_j^2 \lfloor x_0 \rfloor}{2}}. \tag{14}$$

4. We will now bound $P(T_j = t | \bar{X}_{jt} \geq \mu_j + \frac{\Delta_j}{2})$ with $P(T_j^R \leq t)$, which is the probability that arm j has been pulled up to t times at random (the conditioning was dropped since random pulls are independent of the values of the averages). The sum can then be bounded with

$$\sum_{t=1}^{\lfloor x_0 \rfloor} P(T_j = t | \bar{X}_{jt} \geq \mu_j + \frac{\Delta_j}{2}) \leq \sum_{t=1}^{\lfloor x_0 \rfloor} P(T_j^R \leq t) \leq x_0 P(T_j^R \leq x_0). \tag{15}$$

Now, we need a bound for $P(T_j^R \leq x_0)$. Because T_j^R is the sum of binary random variables, we can use the Bernstein inequality to obtain a bound. Consider random variables $A_1, \ldots, A_n$ with range $[0, 1]$ and $\sum_{i=1}^{n} \mathrm{Var}[A_i | A_{i-1}, \ldots, A_1] = \sigma^2$. The Bernstein inequality is

$$P(\sum_{i=1}^{n} A_i \leq \sum_{i=1}^{n} E[A_i] - a) \leq e^{-\frac{a^2/2}{\sigma^2 + a/2}} \tag{16}$$

for any $a \geq 0$.

Show that choosing $x_0 = \frac{1}{2} E[T_j^R]$ yields the following exponential bound:

$$P(T_j^R \leq x_0) \leq e^{-\frac{x_0}{5}}. \tag{17}$$

5. Putting everything together we have

$$P(I_n = j) \leq \frac{\epsilon_n}{K} + 2x_0 e^{-\frac{x_0}{5}} + \frac{4}{\Delta_j^2} e^{\frac{-\Delta_j^2 \lfloor x_0 \rfloor}{2}}. \tag{18}$$

The last thing to do is to find a lower bound for x_0 in terms of n. Recall that $x_0 = \frac{1}{2K} \sum_{t=1}^{n} \epsilon_t$. From the definition of the algorithm, $\epsilon_t = 1$ when $t < \frac{cK}{d^2}$ and $\epsilon_t = \frac{cK}{d^2 t}$ otherwise. Prove the following lower bound on x_0:

$$x_0 \geq \frac{c}{d^2} \ln \frac{(n-1)d^2 e^{1/2}}{cK}. \tag{19}$$

6. Substitute the lower bound for x_0 into eq. 18 and show that this implies

$$P(I_n = j) \leq \frac{\epsilon_n}{K} + o(\frac{1}{n}). \tag{20}$$

Comment on any conditions on the parameters c and d that are necessary for this result to hold.

7. Why does this result (i.e., $P(I_n = j) \leq \frac{\epsilon_n}{K} + o(\frac{1}{n})$) imply that the regret of the ϵ_n-Greedy algorithm grows as $O(K \log n)$?