A-Learning for Approximate Planning

D. Blatt
Department of Electrical Engineering
and Computer Science
University of Michigan
Ann Arbor, MI 48109-2122
dblatt@umich.edu

S. A. Murphy[*]
Department of Statistics
University of Michigan
Ann Arbor, MI 48109-1092
samurphy@umich.edu

J. Zhu
Department of Statistics
University of Michigan
Ann Arbor, MI 48109-1092
jizhu@umich.edu

Abstract

Abstract We consider a new algorithm for reinforcement learning called A-learning. A-learning learns the advantages from a single training set. We compare A-learning with function approximation to Q-learning with function approximation and find that because A-learning approximates only the advantages it is less likely to exhibit bias due to the function approximation as compared to Q-learning.

1 Introduction

Recently researchers in the medical and social sciences have become interested in estimating decision policies in settings in which the system dynamics are poorly understood but for which a training set of finite horizon trajectories is available for learning and planning [1]-[4]. In particular a method we call A-learning was proposed by Murphy [3] and further developed by Robins [4] for the purpose of estimating a good decision policy in this setting. A challenge that frequently arises in these settings is that few variables in the high dimensional observed history are important for choosing among possible actions yet many of these variables are important for predicting the next state or reward following an action. This empirical observation has resulted in several principles. First the Hierarchical Ordering Principle and second, the Effect Heredity Principle [5]; taken together these two principles imply that in data analysis, variables are more likely to contribute to main effects (prediction of next state) as opposed to interactions (used in choosing among potential actions). A-learning accommodates this challenge by estimating only the advantage (the Q-function minus its maximal value [6]) as opposed to estimating the entire Q-function.

In this paper, we introduce the A-learning algorithm with function approximation. We compare A-learning to Q-learning in two simple examples. The first example is artificial

[*] Contact author, http://www.stat.lsa.umich.edu/ samurphy/

but illustrative as we use a generative model that allows us to specify the form of the optimal advantages; by varying these specifications we gain intuition into the relative strengths of these two learning algorithms. The second example is a simple partially observed Markov decision process (POMDP) in which neither an explicit form of the optimal Q-function (as a function of the observable history) or the optimal advantage is available.

2 Preliminaries

We begin with some standard definitions for a finite horizon, possibly non-Markovian, decision process. In the following we use upper case letters, O and A to denote random variables and lower case letters, o and a to denote instantiates or values of the random variables. The training set is composed of n finite horizon trajectories, each trajectory is of the form $\{o_0,\ a_0,\ r_0,\ o_1,\ \ldots,\ a_K,\ r_K,\ o_{K+1}\}$ where K is a finite constant. Define $\mathbf{o}_t = \{o_0, \ldots, o_t\}$ and similarly for $\mathbf{a}_t$. Each action A_t takes values in finite, discrete action space $\mathcal{A}$ and O_t takes values in the observation space $\mathcal{O}$. The observation space may be continuous. The rewards are $r_t(\mathbf{o}_t, \mathbf{a}_t, o_{t+1})$ for each $0 \leq t \leq K$ (if the Markov assumption holds then replace $\mathbf{o}_t$ with o_t and $\mathbf{a}_t$ with a_t).

We assume each trajectory in the training set is distributed according to a fixed distribution; in part this distribution depends on the exploration policy. We denote the exploration policy by $\mathbf{p}_K = \{p_0, \ldots, p_K\}$ where the probability that action a is taken given history $\{\mathbf{o}_t, \mathbf{a}_{t-1}\}$ is $p_t(a|\mathbf{o}_t, \mathbf{a}_{t-1})$ (if the Markov assumption holds then as before replace $\mathbf{o}_t$ with o_t and $\mathbf{a}_{t-1}$ with a_{t-1}.) We assume that $p_t(a|\mathbf{o}_t, \mathbf{a}_{t-1}) > 0$ for each action $a \in \mathcal{A}$ and for each possible value $(\mathbf{o}_t, \mathbf{a}_{t-1})$; that is at each time all actions are possible. Denote expectations with respect to the distribution of a trajectory generated under this exploration policy by an E.

Define a deterministic, but possibly non-stationary and non-Markovian policy, π as a sequence of decision rules, $\{\pi_0, \ldots, \pi_K\}$ where the time t decision rule is a mapping from the observable history $(\mathbf{o}_t, \mathbf{a}_{t-1})$ to the action space, e.g. $\pi_t(\mathbf{o}_t, \mathbf{a}_{t-1})$ is an action. Denote expectations with respect to the distribution of a trajectory generated under policy π by an E_π.

Let Π be a collection of all policies (permitting non-stationary, non-Markovian policies). We seek to estimate a policy that maximizes $E_\pi[\sum_{j=0}^{K} r_j(\mathbf{O}_j, \mathbf{A}_j, O_{j+1})|O_0 = o_0]$ over $\pi \in \Pi$. In the Markovian setting, this finite horizon problem provides an approximation to the discounted infinite horizon problem when K is large [7], each policy in Π is composed of memoryless decision rules and for γ a discount factor, each $r_j(\mathbf{o}_j, \mathbf{a}_j, o_{j+1})$ is replaced by $\gamma^j r(o_j, a_j, o_{j+1})$ for r a bounded reward function.

The optimal value function, $V^*(o_0)$ for an observation o_0 is

$$V^*(o_0) = \max_{\pi \in \Pi} E_\pi \left[\sum_{j=0}^{K} r_j(\mathbf{O}_j, \mathbf{A}_j, O_{j+1}) \middle| O_0 = o_0 \right]$$

and the optimal t-value function for history $(\mathbf{o}_t, \mathbf{a}_{t-1})$ is

$$V_t^*(\mathbf{o}_t, \mathbf{a}_{t-1}) = \max_{\pi \in \Pi} E_\pi \left[\sum_{j=t}^{K} r_j(\mathbf{O}_j, \mathbf{A}_j, O_{j+1}) \middle| \mathbf{O}_t = \mathbf{o}_t, \mathbf{A}_{t-1} = \mathbf{a}_{t-1} \right].$$

As is well-known the optimal value functions satisfy the Bellman equations

$$V_t^*(\mathbf{o}_t, \mathbf{a}_{t-1}) = \max_{a_t \in \mathcal{A}} E[r_t(\mathbf{O}_t, \mathbf{A}_t, O_{t+1}) + V_{t+1}^*(\mathbf{O}_{t+1}, \mathbf{A}_t)|\mathbf{O}_t = \mathbf{o}_t, \mathbf{A}_t = \mathbf{a}_t].$$

Optimal, deterministic, time t decision rules must satisfy,

$$\pi_t^*(\mathbf{o}_t, \mathbf{a}_{t-1}) \in \arg\max_{a_t \in \mathcal{A}} E[r_t(\mathbf{O}_t, \mathbf{A}_t, O_{t+1}) + V_{t+1}^*(\mathbf{O}_{t+1}, \mathbf{A}_t) | \mathbf{O}_t = \mathbf{o}_t, \mathbf{A}_t = \mathbf{a}_t].$$

The optimal time t Q-function for policy π is

$$Q_t^*(\mathbf{o}_t, \mathbf{a}_t) = E[r_t(\mathbf{O}_t, \mathbf{A}_t, O_{t+1}) + V_{t+1}^*(\mathbf{O}_{t+1}, \mathbf{A}_t) | \mathbf{O}_t = \mathbf{o}_t, \mathbf{A}_t = \mathbf{a}_t],$$

and thus the optimal time t advantage which is given by

$$\mu_t^*(\mathbf{o}_t, \mathbf{a}_t) = Q_t^*(\mathbf{o}_t, \mathbf{a}_t) - V_t^*(\mathbf{o}_t, \mathbf{a}_{t-1})$$

is always nonpositive and furthermore it is maximized in a_t at $a_t = \pi_t^*(\mathbf{o}_t, \mathbf{a}_{t-1})$. The advantage can be interpreted as the gain in performance obtained by following action a_t at time t and thereafter policy π^* as compared to following policy π^* from time t on.

3 A-learning and Q-learning with function approximation on a training set

To enhance the clarity of our A-learning exposition we first review Q-learning [8] as used with a single training set.

3.1 Q-learning

In Q-learning with function approximation we estimate the Q-function using an approximator (i.e. neural networks, decision-trees etc.) [8] and then derive the estimated policy as the argument of the maximum of the estimated Q-function. Denote $\mathcal{Q}_t$ to be the approximation space for the tth Q-function, e.g. $\mathcal{Q}_t = \{Q_t(\mathbf{o}_t, \mathbf{a}_t; \theta) : \theta \in \Theta\}$; θ is a vector of parameters taking values in a parameter space Θ which is a subset of a Euclidean space. Write $\mathbf{E}_n f$ for the average of f over the training set of n trajectories. For convenience set Q_{K+1} equal to zero. We use dynamic programming and permit the estimator to have different parameters for each time t. Below we abbreviate $r_t(\mathbf{O}_t, \mathbf{A}_t, O_{t+1})$ by r_t.

Q-Learning For t=K, K-1,. . . , 0, set

$$\hat{Y}_t \leftarrow r_t + \max_{a_{t+1}} Q_{t+1}(\mathbf{O}_{t+1}, \mathbf{A}_t, a_{t+1}; \theta_{t+1}).$$

Then,

$$\theta_t \leftarrow \arg\min_{\theta} \mathbf{E}_n \left[\hat{Y}_t - Q_t(\mathbf{O}_t, \mathbf{A}_t; \theta)\right]^2. \tag{1}$$

The estimated policy, $\hat{\pi}_Q$ satisfies $\hat{\pi}_{Q,t}(\mathbf{o}_t, \mathbf{a}_{t-1}) \in \arg\max_{a_t} Q_t(\mathbf{o}_t, \mathbf{a}_t; \theta_t)$ for each t.

As discussed by Sutton and Barto, [8] the intuition underlying the minimization of the quadratic form in Q-learning stems from the Bellman equation:

$$E\left[r_t + \max_{a_{t+1}} Q_{t+1}^*(\mathbf{O}_{t+1}, \mathbf{A}_t, a_{t+1}) \Big| \mathbf{O}_t, \mathbf{A}_t\right] = Q_t^*(\mathbf{O}_t, \mathbf{A}_t).$$

If $Q_{t+1} = Q_{t+1}^*$ and the training set were infinite, then the quadratic form in (1) becomes,

$$E\left[r_t + \max_{a_{t+1}} Q_{t+1}^*(\mathbf{O}_{t+1}, \mathbf{A}_t, a_{t+1}) - Q_t(\mathbf{O}_t, \mathbf{A}_t; \theta)\right]^2.$$

Minimizing this is equivalent to minimizing

$$E\left[Q_t^* - Q_t(\mathbf{O}_t, \mathbf{A}_t; \theta)\right]^2.$$

3.2 A-learning

In A-learning [3]-[4] with function approximation we estimate the advantages using an approximator and derive the estimated policy as the argument of the maximum of each estimated advantage. Define $\mathcal{M}_t$ as the approximation space for the tth advantage, e.g. $\mathcal{M}_t = \{\mu_t(\mathbf{o}_j, \mathbf{a}_j; \theta) : \theta \in \Theta\}$; θ is a vector of parameters taking values in a parameter space Θ which is a subset of a Euclidean space. As with Q-learning, we use dynamic programming and permit the estimator to have different parameters for each time t.

A-Learning For t=K, K-1,..., 0, set

$$\hat{Y}_t \leftarrow \sum_{i=t}^{K} r_i - \sum_{i=t+1}^{K} \mu_i(\mathbf{O}_i, \mathbf{A}_i; \theta_i).$$

Then,

$$\begin{aligned} \theta_t &\leftarrow \arg\min_{\theta} \mathbf{E}_n \left[\hat{Y}_t - \mu_t(\mathbf{O}_t, \mathbf{A}_t; \theta) + E(\mu_t(\mathbf{O}_t, \mathbf{A}_t; \theta)|\mathbf{O}_t, \mathbf{A}_{t-1})\right]^2 \\ \mu_t(\mathbf{O}_t, \mathbf{A}_t; \theta_t) &\leftarrow \mu_t(\mathbf{O}_t, \mathbf{A}_t; \theta_t) - \max_{a_t} \mu_t(\mathbf{O}_t, \mathbf{A}_{t-1}, a_t; \theta_t) \qquad (2) \end{aligned}$$

where $E(\mu_t(\mathbf{O}_t, \mathbf{A}_t; \theta)|\mathbf{O}_t, \mathbf{A}_{t-1}) = \sum_{a_t} \mu_t(\mathbf{O}_t, \mathbf{A}_{t-1}, a_t; \theta) p_t(a_t|\mathbf{O}_t, \mathbf{A}_{t-1})$ and the last assignment is simply to ensure that the estimated μ_t is an advantage function. In contrast to Q-learning, A-learning explicitly uses the exploration policy $\{p_0, \ldots, p_K\}$ in the algorithm. The estimated policy, $\hat{\pi}_A$ satisfies $\hat{\pi}_{A,t}(\mathbf{o}_t, \mathbf{a}_{t-1}) \in \arg\max_{a_t} \mu_t(\mathbf{o}_t, \mathbf{a}_t; \theta_t)$ for each t.

The intuition underlying the minimization of this quadratic form (2) is slightly more complicated than that in Q-learning in that this minimization projects $\hat{Y}_t$ on a mean zero space rather than $\mathcal{Q}_t$. First

$$E\left[\sum_{i=t}^{K} r_i - \sum_{i=t+1}^{K} \mu_i^*(\mathbf{O}_i, \mathbf{A}_i)\Big|\mathbf{O}_t, \mathbf{A}_t\right] = Q_t^*(\mathbf{O}_t, \mathbf{A}_t).$$

Note if $(\mu_{t+1}, \ldots, \mu_K) = (\mu_{t+1}^*, \ldots, \mu_K^*)$ and the training set were infinite then the quadratic form above (2) becomes

$$E\left(\sum_{i=t}^{K} r_i - \sum_{i=t+1}^{K} \mu_i^* - \mu_t(\mathbf{O}_t, \mathbf{A}_t; \theta) + E[\mu_t(\mathbf{O}_t, \mathbf{A}_t; \theta)|\mathbf{O}_t, \mathbf{A}_{t-1}]\right)^2.$$

Minimizing this is equivalent to minimizing

$$E\left(Q_t^* - \mu_t(\mathbf{O}_t, \mathbf{A}_t; \theta) + E[\mu_t(\mathbf{O}_t, \mathbf{A}_t; \theta)|\mathbf{O}_t, \mathbf{A}_{t-1}]\right)^2.$$

In turn this is equivalent to minimizing

$$E\Big(Q_t^* - E[Q_t^*(\mathbf{O}_t, \mathbf{A}_{t-1}, a_t)|\mathbf{O}_t, \mathbf{A}_{t-1}] - \mu_t(\mathbf{O}_t, \mathbf{A}_t; \theta) + E[\mu_t(\mathbf{O}_t, \mathbf{A}_t; \theta)|\mathbf{O}_t, \mathbf{A}_{t-1}]\Big)^2.$$

But $Q_t^* - E[Q_t^*(\mathbf{O}_t, \mathbf{A}_{t-1}, a_t)|\mathbf{O}_t, \mathbf{A}_{t-1}] = \mu_t^* - E[\mu_t^*(\mathbf{O}_t, \mathbf{A}_{t-1}, a_t)|\mathbf{O}_t, \mathbf{A}_{t-1}]$! The fact that the advantage must satisfy, $\max_{a_t} \mu_t(\mathbf{O}_t, \mathbf{A}_{t-1}, a_t; \theta) = 0$ means that minimizing this quadratic form over all possible advantages yields $\mu_t = \mu_t^*$.

4 Experiments

The A-learning and Q-learning algorithms do not use knowledge of the system dynamics (e.g. the distribution of O_{t+1} given $(\mathbf{O}_t, \mathbf{A}_t)$ or that the decision process is a MDP or

POMDP). However the experiments considered here are simulated using generative models. Furthermore the algorithms will learn only on a training set. A-learning will use knowledge of the exploration policy. There will be no exploitation; that is we will not alter the exploration policy. These experiments use very simple models. The first experiment is primarily illustrative, in that this model does not necessarily correspond to a physical system, rather via this model we are able to generate finite horizon trajectories with arbitrarily complex advantages. This will permit us to more effectively compare and contrast the two learning algorithms. The second experiment is a simple sensor allocation example arising in tracking.

4.1 Illustrative Experiment

Data Generation. To generate the trajectories, we use the following telescoping expression:

$$\begin{aligned} E_{\pi^*}\left[\sum_{t=0}^{K} r_t \Big| \mathbf{O}_K, \mathbf{A}_K\right] &= \sum_{t=0}^{K} Q_t^*(\mathbf{O}_t, \mathbf{A}_t) - V_t^*(\mathbf{O}_t, \mathbf{A}_{t-1}) \\ &\quad + \sum_{t=1}^{K} r_{t-1} + V_t^*(\mathbf{O}_t, \mathbf{A}_{t-1}) - Q_{t-1}^*(\mathbf{O}_{t-1}, \mathbf{A}_{t-1}) \\ &\quad + V_0^*(O_0) \end{aligned} \tag{3}$$

The summands in the first sum are the advantages. The summands in the second sum are temporal-difference errors [8]. Recall that the conditional expectation of the tth temporal-difference error given $(\mathbf{O}_{t-1}, \mathbf{A}_{t-1})$ is zero:

$$E\left[r_{t-1} + V_t^*(\mathbf{O}_t, \mathbf{A}_{t-1})\big|\mathbf{O}_{t-1}, \mathbf{A}_{t-1}\right] = Q_{t-1}^*(\mathbf{O}_{t-1}, \mathbf{A}_{t-1}).$$

Denote the tth temporal-difference error by $\phi_t(\mathbf{O}_t, \mathbf{A}_{t-1})$, $t = 0, \ldots, K$.

For convenience choose the first component of each O_t equal to r_{t-1}. Next choose conditional distributions for each O_t given $(\mathbf{O}_{t-1}, \mathbf{A}_{t-1})$ and exploration policy $\{p_0, \ldots, p_K\}$. Thus the conditional distribution of each reward except r_K has been specified. Choose a mean zero distribution for r_K given $(\mathbf{O}_K, \mathbf{A}_K)$; call this $f_K(\cdot|\mathbf{O}_T, \mathbf{A}_T)$. Choose a form for each of the advantages. For example if the actions are all binary, a possible choice is

$$\mu_t^* = A_t g_t(\mathbf{O}_t, \mathbf{A}_{t-1}) - [g_t(\mathbf{O}_t, \mathbf{A}_{t-1})]^+ \tag{4}$$

where g_t is a specified function of $(\mathbf{O}_t, \mathbf{A}_{t-1})$ and $[x]^+$ is x if $x > 0$ and zero otherwise. Choose a form for $V_0^*(O_0)$. Choose functional forms for each of the temporal difference errors, say $r_{t-1} + \phi_t'(\mathbf{O}_t, \mathbf{A}_{t-1})$; these do not necessarily have mean zero. However during the generation of the trajectory we will subtract the conditional means.

To generate a trajectory first generate O_0, generate A_0 and then generate $O_1 = (r_0, O_1)$ (recall the first component of O_t is r_{t-1}). Generate A_1. Using the previously specified conditional distribution of O_1 given (O_0, A_0), calculate $\phi_1(\mathbf{O}_1, A_0) = r_0 + \phi_1'(\mathbf{O}_1, A_0) - E[r_0 + \phi'|O_0, A_0]$. Now repeat, beginning with the generation of A_1 until A_K. Lastly draw a variable from the distribution $f_K(\cdot|\mathbf{O}_K, \mathbf{A}_K)$ and add

$$\sum_{t=0}^{K} \mu_t^*(\mathbf{O}_t, \mathbf{A}_t) + \sum_{t=1}^{K} (\phi_t(\mathbf{O}_t, \mathbf{A}_{t-1}) - r_{t-1}) + V_0^*(O_0).$$

This sum is r_K. This procedure produces one trajectory $\{O_0, A_0, r_0, O_1, \ldots, O_K, A_K, r_K\}$ drawn from a distribution with the specified advantages [3].

The experiments. We conduct two experiments. In each experiment both Q-learning and A-learning learn on a training set of 1000 trajectories. The resulting policies are then tested on a different set of 10000 trajectories. Lasso [9] combined with Bagging [10] is used in both Q-learning and A-learning. Lasso is a supervised learning technique that fits a L_1-constrained least squares linear regression model. It has the so called *sparse* property that automatically eliminates the irrelevant features and keeps the relevant ones. Note that this means that the approximation space for both the Q-function and advantage is linear in the observable history. Bagging is used for the purpose of reducing the variance of fitted models. In both experiments, the generative models conform to the hierarchical ordering principle and effect heredity principle and thus the advantages depend on a few features and the complexity of the advantage is low; that is the $g_t(\mathbf{O}_t, \mathbf{A}_{t-1})$'s in the definition of the advantages (4) are linear in $(\mathbf{O}_t, \mathbf{A}_{t-1})$.

In experiment A, we first construct each temporal-difference error as a linear function of $(\mathbf{O}_t, \mathbf{A}_{t-1})$. The number of decision epochs is 3 ($K = 2$) and the dimension of each O_t is 5 (hence the dimension of $(\mathbf{O}_t, \mathbf{A}_{t-1})$ is $5(t + 1) + t$), but only 3 components of O_t enter the advantage at time t, the remaining $5(t + 1) + t - 3$ features are noise features. Of course neither Q-learning nor A-learning know how many features are relevant in the advantage. Secondly in experiment A we increase the dimension of each O_t from 5 to 10 (as before only 3 components are relevant in (4) , and thirdly we increase the horizon from 3 decision epochs to 10. Figure 2 (left panel) shows the result, which are boxplots over 20 independently generated training-test combinations. Both A-learning and Q-learning are dramatically better than the random policy that was used to generate the data (not shown in the figure), and as we can see, A-learning is also consistently better than Q-learning in this simple experiment. When the horizon is short, the difference between A-learning and Q-learning is small, but when the horizon increases, the difference between A-learning and Q-learning becomes quite dramatic. It is also interesting to notice that when we increase the dimension of O_t, the performance of Q-learning and A-learning stay about the same. We attribute this to the effectiveness of Lasso in eliminating the irrelevant features.

In experiment B, each temporal-difference error is a nonlinear function of $\mathbf{O}_t$ and $\mathbf{A}_{t-1}$, but $g_t(\mathbf{O}_t, \mathbf{A}_{t-1})$ in the advantage function remains linear in the observable history. Since the approximation space for both Q-learning and A-learning is linear in the observable history, we expect Q-learning will suffer more from the nonlinearity due to the bias than A-learning . As we can see in the right panel of Figure 2, this is indeed the case. As the horizon increases, similar as in the case when the temporal difference error is linear, the advantage of A-learning over Q-learning increases. However, the effect is less pronounced, due to a variety of reasons, not all of which are clear to us at this point. We conjecture it is due to the added variance from the nonlinear temporal difference errors.

4.2 Sensor Allocation

Next we consider the application of Q-learning and A-learning to a sensor allocation problem [11]. In this generative model, the target starts at position $[0, 0]$ on the x-y plane and moves at a known constant speed but with unknown constant angle θ. Therefore, the target's position at time t is $t[\cos(\theta), \sin(\theta)]$, i.e. it is known that at time t the target lies on the circle of radius t but its location on the circle is unknown. The tracking device takes 5 observations at consecutive discrete times. At each observation, one of two sensors can be used. The first sensor is a narrow angle sensor, which covers a circular area with radius $\sqrt{1/2}$. The second is a wide angle sensor which covers a circular area with radius $\sqrt{3/2}$. For simplicity, both sensors have zero misdetect and false alarm probabilities. If the narrow (wide) angle sensor is used and the target lies in the area which it covers, a reward of 1 (0.3) units is granted. The goal is to maximize the total reward at the end of the fifth stage. At every time point, given the past observable history, the center of the sensor's angle is cho-

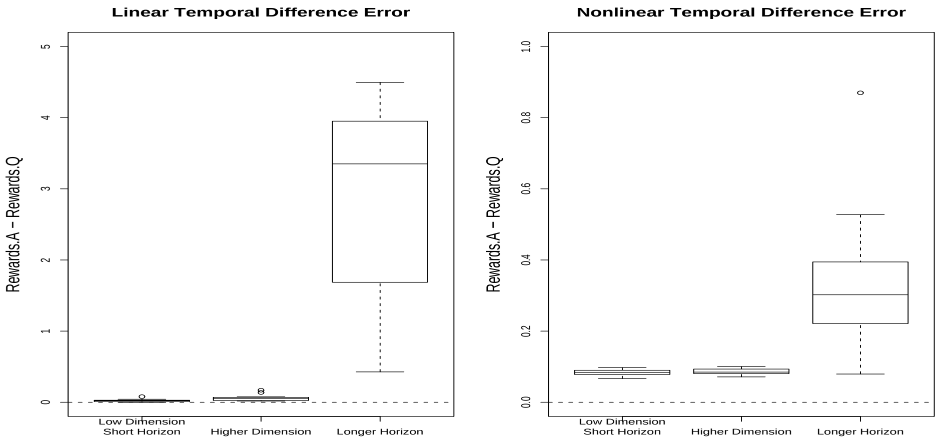

Figure 1: A illustrative comparison of Q-learning and A-learning.

sen at random from the set of angles at which the target can be. We consider this element of the problem fixed. Hence, at each time point, given the past history, the only decision to be made is the type of sensor to use.

In the simulation, 10 training sets each of 2000 trajectories were generated. At each trajectory, θ was randomly selected from $[0, 2\pi)$, and the tracking device selected the sensors at random, giving each sensor equal probability. We did not know which functions of the observable history would be most informative of the optimal policy and when we used Lasso neither algorithm performed well. Thus instead we used Random Forest [12] which uses a more flexible approximation space (this method approximates functions using averages of decision trees, each tree is constructed on a resampled version of the training set). In the simulation, 500 trees, each with terminal nodes of 400 samples, were averaged.

In this setting Q-learning and A-learning have comparable performance, in terms of average total reward, which is better than the performance of the random policy. The results are summarized in Fig. 2. Since the Random Forest approximation is highly flexible (although very computationally intensive) approximation method, there is no reason to suspect that A learning will perform better than Q-learning. Therefore, it is encouraging to see that even in this scenario A-learning is not worse than Q-learning.

5 Discussion

The Hierarchical Ordering and the Effect Heredity Principles imply that in the analysis of data, variables are more likely to contribute to main effects (prediction of next state) as opposed to interactions (used in choosing among potential actions). If these principles hold then our illustrative example suggests that A-learning will perform better than Q-learning. However other experiments, not shown here, and to a lesser extent the sensor allocation experiment discussed above, indicate that A-learning can be highly variable. Variance reduction techniques such as Bagging and Random Forests appear necessary for a successful implementation of A-learning. A-learning is a new reinforcement learning method that shows promise but when it is most useful and its general properties are not fully understood at this time.

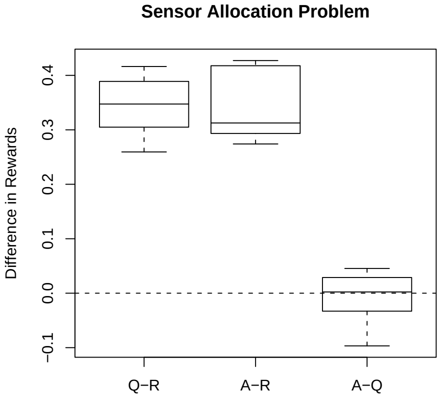

Figure 2: Sensor Allocation Problem.

Acknowledgments

We acknowledge support from NIH P50 DA10075 and K02 DA15674 (Murphy).

References

[1] Schneider LS, PN Tariot, et al. (2001). National Institute of Mental Health Clinical Antipsychotic Trials of Intervention Effectiveness (CATIE). *Am J Geriatr Psychiatry* **9(4)**:346-360.

[2] Fava M, AJ Rush, et al. . (2003). Background and Rationale for the Sequenced Treatment Alternative to Relieve Depression (STAR*D) Study. *Psychiatric Clinics of North America* **26(3)**:457-494.

[3] Murphy S.A. (2003), Optimal Dynamic Treatment Regimes. *Journal of the Royal Statistical Society, Series B (with discussion)* **65**(2):331-366.

[4] Robins J. (2003). Optimal Structural Nested Models. *unpublished manuscript.*

[5] Wu J.C.F & M. Hamada (2000) *Experiments: Planning, Analysis, and Parameter Design Optimization*. New York: John Wiley & Sons, Inc.

[6] Baird LC. (1993). Advantage Updating. *Wright Lab. Technical Report.* WL-TR-1146.

[7] Kearns M, Y Mansour & AY Ng (2000). Approximate planning in large POMDPs via reusable trajectories. In *Neural Information Processing Systems* 12 MIT Press.

[8] Sutton RS & AG Barto (1998), *Reinforcement Learning: An Introduction*. Cambridge, Mass.: The MIT Press.

[9] Tibshirani R (1996) Regression shrinkage and selection via the Lasso. *Journal of Royal Statistical Society B*, Vol. 58, No. 1.

[10] Breiman L (1996) Bagging predictors. *Machine Learning*, 26, 123-140.

[11] Kreucher, C., K. Castella, & A. O. Hero (2003). Multi-target sensor management using alpha divergence measures, in *Proc. First IEEE Conference on Information Processing in Sensor Networks*, Palo Alto.

[12] Breiman L (2001) Random Forests *Machine Learning*, 45, 5-32.