massachusetts institute of technology — artificial intelligence laboratory

Policy Improvement for POMDPs Using Normalized Importance Sampling

Christian R. Shelton

AI Memo 2001-002 March 20, 2001

Abstract

We present a new method for estimating the expected return of a POMDP from experience. The estimator does not assume any knowledge of the POMDP and allows the experience to be gathered with an arbitrary set of policies. The return is estimated for any new policy of the POMDP. We motivate the estimator from function-approximation and importance sampling points-of-view and derive its theoretical properties. Although the estimator is biased, it has low variance and the bias is often irrelevant when the estimator is used for pair-wise comparisons. We conclude by extending the estimator to policies with memory and compare its performance in a greedy search algorithm to the REINFORCE algorithm showing an order of magnitude reduction in the number of trials required.

This report describes research done within CBCL in the Department of Brain and Cognitive Sciences and in the AI Lab at MIT. This research is sponsored by a grants from ONR contracts Nos. N00014-93-1-3085 & N00014-95-1-0600, and NSF contracts Nos. IIS-9800032 & DMS-9872936. Additional support was provided by: AT&T, Central Research Institute of Electric Power Industry, Eastman Kodak Company, Daimler-Chrysler, Digital Equipment Corporation, Honda R&D Co., Ltd., NEC Fund, Nippon Telegraph & Telephone, and Siemens Corporate Research, Inc.

1 Introduction

We assume a standard reinforcement learning setup: an agent interacts with an environment modeled as a partially-observable Markov decision process. Consider the situation after a sequence of interactions. The agent has now accumulated data and would like to use that data to select how it will act next. In particular, it has accumulated a sequence of observations, actions, and rewards and it would like to select a policy, a mapping from observations to actions, for future interaction with the world. Ultimately, the goal of the agent is to find a policy mapping that maximizes the agent's return, the sum of rewards experienced.

[3] presents a method for estimating the return for every policy simultaneously using data gathered while executing a fixed policy. In this paper we consider the case where we do not restrict the policies used for gathering data. Either we did not have control over the method for data collection, or we would like to allow the learning algorithm the freedom to pick any policy for any trial and still be able to use the data.

In the next section we develop two estimators (unnormalized and normalized). Section 3 shows that while the normalized estimator is biased, its variance is much lower than the unnormalized (unbiased) estimator resulting in a better estimator for comparisons. Section 4 demonstrates some results on simulated environments. We conclude with a discussion of how to improve the estimator further.

2 Estimators

2.1 Notation

In this paper we will use s to represent the hidden state of the world, x for the observation, a for the action, and r for the reward. Subscripts denote the time index and superscripts the trial number. We assume episodes of fixed-length, T.

Let $\pi(x, a)$ be a policy (the probability of picking action a upon observing x). For the moment we will consider only reactive policies of this form. h represents a history[1] (of T time steps) and therefore is a tuple of four sequences: states (s_1 through s_T), observations (x_1 through x_T), actions (a_1 through a_T), and rewards (r_1 through r_T). The state sequence is not available to the algorithm and is for theoretical consideration only. Lastly, we let R be the return (or sum of r_1 through r_T).

π^1 through π^n are the n policies tried. h^1 through h^n are the associated n histories with R^1 through R^n being the returns of those histories. Thus during trial i the agent executed policy π^i resulting in the history h^i. R^i is used as a shorthand notation for $R(h^i)$, the return of that history.

2.2 Importance Sampling

Importance sampling is typically presented as a method for reducing the variance of the estimate of an expectation by carefully choosing a sampling distribution[9]. For example, the most direct method for evaluating $\int f(x)p(x)\,dx$ is to sample i.i.d. $x_i \sim p(x)$ and use $\frac{1}{n}\sum_i x_i$ as the estimate. However, by choosing a different distribution $q(x)$ which has higher density in the places where $|f(x)|$ is larger, we can get a new estimate which is still unbiased and has lower variance. In particular, we now draw $x_i \sim q(x)$ and use $\frac{1}{n}\sum_i x_i \frac{p(x_i)}{q(x_i)}$ as our estimate. This can be viewed as estimating the expectation of $f(x)\frac{p(x)}{q(x)}$ with respect to $q(x)$ which is like approximating $\int f(x)\frac{p(x)}{q(x)}q(x)\,dx$ with samples drawn from $q(x)$. If $q(x)$ is chosen properly, our new estimate has lower variance.

[1] It might be better to refer to this as a trajectory since we will not limit h to represent only sequences that have been observed; it can also stand for sequences that might be observed. However, the symbol t is over used already. Therefore, we have chosen to use h to represent state-observation-action-reward sequences.

In this paper, instead of choosing $q(x)$ to reduce variance, we will be forced to use $q(x)$ because of how our data was collected. Instead of the traditional setting where an estimator is chosen and then a distribution is derived which will achieve minimal variance, we instead have a distribution chosen and we are trying to find an estimator with low variance.

2.3 Sampling Ratios

We have accumulated a set of histories (h^1 through h^n) each recorded by executing a (possibly different) policy (π^1 through π^n). We would like to use this data to find a guess at the best policy.

A key observation is that we can calculate one factor in the probability of a history given a policy. In particular, that probability has the form

$$\begin{aligned} p(h|\pi) &= p(s_1) \prod_{t=1}^{T} p(x_t|s_t)\pi(x_t, a_t)p(s_{t+1}|s_t, a_t) \\ &= \left[p(s_1) \prod_{t=1}^{T} p(x_t|s_t)p(s_{t+1}|s_t, a_t) \right] \left[\prod_{t=1}^{T} \pi(x_t, a_t) \right] \\ &= W(h)A(h, \pi) \ . \end{aligned}$$

$A(h, \pi)$, the effect of the agent, is computable whereas $W(h)$, the effect of the world, is not because it depends on knowledge of the underlying state sequence. However, $W(h)$ does not depend on π. This implies that the ratios necessary for importance sampling are exactly the ratios that are computable without knowing the state sequence. In particular, if a history h was drawn according to the distribution induced by π and we would like an unbiased estimate of the return of π', then we can use $R(h)\frac{p(h|\pi')}{p(h|\pi)}$ and although neither the numerator nor the denominator of the importance sampling ratio can be computed, the $W(h)$ term in each cancels leaving a ratio of $A(h, \pi')$ to $A(h, \pi)$ which can be calculated. Different statements of the same fact have been shown before in [5, 7]. This fact will be exploited in each of the estimators in this paper.

2.4 Importance Sampling as Function Approximation

Because each π^i is potentially different, each h^i is drawn according to a different distribution and so while the data are drawn independently, they are not identically distributed. This makes it difficult to apply importance sampling directly. The most obvious thing to do is to construct n estimators (one from each data point) and then average them. If π is the policy we are trying to evaluate, this estimator is

$$\frac{1}{n} \sum_i R^i \frac{p(h^i|\pi)}{p(h^i|\pi^i)} \ . \tag{1}$$

This estimator has the problem that its variance can be quite high. In particular, if only one of the sampled policies is close to the target policy, then only one of the elements in the sum will have a low variance. The other variances will be very high and overwhelm the total estimate. We might then only use the estimate from the policy that is most similar to the target policy. Yet, we would hope to do better by using all of the data.

However, we can use all of the data and reduce the variance if we consider importance sampling from a function approximation point-of-view. Importance sampling in general seeks to estimate $\int f(x)p(x)\,dx$. Consider estimating this integral by evaluating $\int \hat{f}(x)\hat{p}(x)\,dx$ where $\hat{f}$ and $\hat{p}$ are

approximations of f and p derived from data. In particular, with a bit of foresight we will choose $\hat{f}$ and $\hat{p}$ to be nearest-neighbor estimates. Let $i(x)$ be the index of the data point nearest to x. Then,

$$\hat{f}(x) = f(x^{i(x)})$$
$$\hat{p}(x) = p(x^{i(x)}) \ .$$

Letting α_i be the size of the region of the observation space closest to x^i,

$$\int \hat{f}(x)\hat{p}(x)\,dx = \sum_i \alpha_i f(x^i)p(x^i) \ .$$

We will also need to approximate α_i. Let $q(x)$ be the distribution from which x^i was sampled. We will take the estimate of α_i to be proportional to the reciprocal of the density, $\frac{1}{q(x^i)}$. This yields the standard importance sampling estimator

$$\frac{1}{n}\sum_i f(x^i)\frac{p(x^i)}{q(x^i)} \ .$$

More importantly, this derivation gives insight into how to merge samples from different distributions. We can use the same approximations for $\hat{f}$ and $\hat{p}$. The only difference comes in the approximation for α_i. Whereas before the density of samples was $q(x)$, now if each sample was drawn according to its own distribution, $q^i(x)$, then the density is $\frac{1}{n}\sum_i q^i(x)$. Applying this change results in the estimator

$$\sum_i f(x^i)\frac{p(x^i)}{\sum_j q^j(x^i)} \ .$$

which, when translated to the POMDP estimation problem becomes

$$\sum_{i=1}^{n} R^i \frac{p(h^i|\pi)}{\sum_{j=1}^{n} p(h^i|\pi^j)} \ . \tag{2}$$

This estimator is also unbiased (shown in the appendix) and has a lower variance because if one of the sampling distributions is near the target distribution, then all elements in the sum share the benefit.

2.5 Normalized Estimates

We can normalize the importance sampling estimate to obtain a lower variance estimate at the cost of adding bias. Previous work has used a variety of names for this including weighted uniform sampling[9], weighted importance sampling[7], and ratio estimate[2]. In general, such an estimator has the form

$$\frac{\sum_i f(x^i)\frac{p(x^i)}{q(x^i)}}{\sum_i \frac{p(x^i)}{q(x^i)}} \ .$$

This new form can be viewed in three different ways. First, it can be seen just as a trick to reduce variance. Second, it can be viewed as a Bayesian estimate of the expectation [1, 4]. Unfortunately, this view does not work for our application because we do not know the true probabilities densities. [2] connects the ratio and Bayesian views, but neither can be applied here.

The final view is that we have adjusted the function approximator $\hat{p}$. The problem with the previous estimator can be seen by noting that the function approximator $\overline{p}(h)$ does not integrate

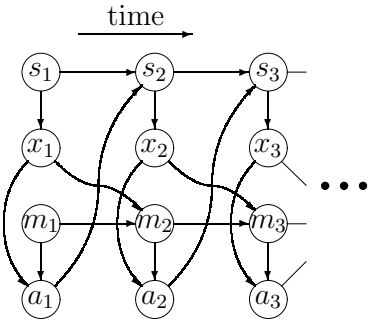


Figure 1: Dependency graph for agent-world interaction with memory model

(or sum) to 1. Instead of $\hat{p} = p(x^{i(x)})$, we make sure $\hat{p}$ integrates (or sums) to 1: $\hat{p} = p(x^{i(x)})/Z$ where $Z = \sum_i \alpha_i p(x^i)$. When recast in terms of our POMDP problem the estimator is

$$\frac{\sum_{i=1}^{n} R^i \frac{p(h^i|\pi)}{\sum_{j=1}^{n} p(h^i|\pi^j)}}{\sum_{i=1}^{n} \frac{p(h^i|\pi)}{\sum_{j=1}^{n} p(h^i|\pi^j)}} \ . \tag{3}$$

2.6 Adding Memory

So far we have only discussed estimators for reactive policies (policies that map the immediate observation to an action). We would like to be able to also estimate the return for policies with memory. Consider adding memory in the style of a finite-state controller. At each time step, the agent reads the value of the memory along with the observation and makes a choice about which action to take and the new setting for the memory. The policy now expands to the form $\pi(x, m, a, m') = p(a, m'|x, m)$, the probability of picking action a and new memory state m' given observation x and old memory state m. Now let us factor this distribution, thereby limiting the class of policies realizable by a given memory size slightly but making the model simpler. In particular we consider an agent model where the agent's policy has two parts: $\pi_a(x, m, a)$ and $\pi_m(x, m, m')$. The former is the probability of choosing action a given that the observation is x and the internal memory is m. The latter is the probability of changing the internal memory to m' given the observation is x and the internal memory is m. Thus $p(a, m'|x, m) = \pi_a(x, m, a)\pi_m(x, m, m')$. By this factoring of the probability distribution of action-memory choices, we induce the dependency graph shown in figure 1.

If we let M be the sequence $\{m_1, m_2, \ldots, m_T\}$, $p(h|\pi)$ can be written as

$$\begin{aligned}
\sum_M p(h, M|\pi) &= \sum_M p(s_1)p(m_1) \prod_{t=1}^{T} p(x_t|s_t)\pi_a(x_t, m_t, a_t)\pi_m(x_t, m_t, m_{t+1})p(s_{t+1}|s_t, a_t) \\
&= \left[p(s_1) \prod_{t=1}^{T} p(x_t|s_t)p(s_{t+1}|s_t, a_t) \right] \left[\sum_M p(m_1) \prod_{t=1}^{T} \pi_a(x_t, m_t, a_t)\pi_m(x_t, m_t, m_{t+1}) \right] \\
&= W(h)A(h, \pi) \ ,
\end{aligned}$$

once again splitting the probability into two parts: one for the world dynamics and one for the agent dynamics. The $A(h, \pi)$ term involves a sum over all possible memory sequences. This can

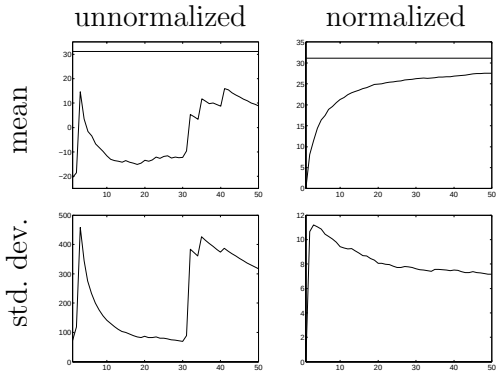


Figure 2: Means and standard deviations for the unnormalized and normalized estimates of the return differences as a function of the number of data points. The data were collected by executing the policy corresponding to the point $(0.4, 0.6)$ in figure 3. The estimators were evaluated at the policies $(0.3, 0.9)$ and $(0.4, 0.5)$. Plotted above are the means and standard deviations of these estimates over 600 experiments. The horizontal line on the plots of the mean represent the true difference in returns. The normalized plots fit the theoretical values well. The unnormalized plots demonstrate that the variance is large enough that even 600 samples are not enough to get a good estimate. The unnormalized mean should be constant at the true return difference and the standard deviation should decay as $\frac{1}{\sqrt{n}}$. However, because the unnormalized estimator is much more asymmetric (it relies on a few very heavily weighted unlikely events to offset the more common events), the graph does not correspond well to the theoretical values. This is indicative of the general problem with the high variance unnormalized estimates.

easily be computed by noting that $A(h, \pi)$ is the probability of the action sequence given the observation sequence where the memory sequence is unobserved. This is a (slight) variation of a hidden Markov model: an input-output HMM. The difference is that the HMM transition and observation probabilities for a time step (the memory policy and the action policy respectively) depend on the value of x at that time step. Yet, the x's are visible making it possible to compute the probability and its derivative by using the forward-backward algorithm.

As such, we can now use the same estimators and allow for policies with memory. In particular, the estimator has explicit knowledge of the working of the memory. This is in direct contrast to the method of adding the memory to the action and observation spaces and running a standard reinforcement learning algorithm where the agent must learn dynamics of its own memory. With our explicit memory model, the learning algorithm understands that the goal is to produce the correct action sequence and uses the memory state to do so by coordinating the actions in different time steps.

3 Estimator Properties

It is well known that importance sampling estimates (both normalized and unnormalized) are consistent[2, 1, 4]. Additionally, normalized estimators have smaller asymptotic variance if the sampling distribution does not exactly match the distribution to estimate[2]. However, we are more interested in the case of finite sample sizes.

The estimators of section 2.4 (equations 1 and 2) are unbiased. That is, for a set of chosen policies, $\{\pi^1, \pi^2, \ldots, \pi^n\}$, the expectation over experiences of the estimator evaluated at π is the

true expected return for executing policy π. Similarly, the estimator of section 2.5 (equation 3) is biased. In specific, it is biased towards the expected returns of $\{\pi^1, \pi^2, \ldots, \pi^n\}$.

The goal of constructing these estimators is to use them to choose a good policy. This involves comparing the estimates for different values of π. Therefore instead of considering a single point we will consider the difference of the estimator evaluated at two different points, π_A and π_B. In other words, we will use the estimators to calculate an estimate of the difference in expected returns between two policies. The appendix details the derivation of the biases and variances. We only quote the results here. These results are for using the same data for both the estimate at π_A and the estimate at π_B.

We will consider only the unnormalized estimator (equation 2) and the normalized estimator (equation 3), denoting them as D_U and D_N respectively. First, a few useful definitions:

$$
\begin{aligned}
\overline{p}(h) &= \frac{1}{n}\sum_i p(h|\pi^i) \\
\tilde{p}(h,g) &= \frac{1}{n}\sum_i p(h|\pi^i)p(g|\pi^i) \\
s^2_{X,Y} &= \int R^2(h)\frac{p(h|\pi_X)p(h|\pi_Y)}{\overline{p}(h)}\,dh \\
\overline{s^2_{X,Y}} &= \int (R(h)-E[R|\pi_X])(R(h)-E[R|\pi_Y])\frac{p(h|\pi_X)p(h|\pi_Y)}{\overline{p}(h)}\,dh \\
\eta^2_{X,Y} &= \iint R(h)R(g)\frac{p(h|\pi_X)p(g|\pi_Y)}{\overline{p}(h)\overline{p}(g)}\tilde{p}(h,g)\,dh\,dg \\
\overline{\eta^2_{X,Y}} &= \iint (R(h)-E[R|\pi_X])(R(g)-E[R|\pi_Y])\frac{p(h|\pi_X)p(g|\pi_Y)}{\overline{p}(h)\overline{p}(g)}\tilde{p}(h,g)\,dh\,dg \\
b_{A,B} &= \iint [R(h)-R(g)]\frac{p(h|\pi_A)p(g|\pi_B)}{\overline{p}(h)\overline{p}(g)}\tilde{p}(h,g)\,dh\,dg\ .
\end{aligned}
$$

Note that all of these quantities are invariant to the number of samples provided that the relative frequencies of the sampled policies remain the same as the number of samples increases. $\overline{p}$ and $\tilde{p}$ are measures of the average distribution over histories. $s^2_{X,Y}$ and $\eta^2_{X,Y}$ are measures of second moments and $\overline{s^2_{X,Y}}$ and $\overline{\eta^2_{X,Y}}$ are measures of (approximately) centralized second moments. $b_{A,B}$ is the bias of the normalized estimate of the return difference.

The biases and variances of the estimates are[2]

$$
\begin{aligned}
E[D_U] &= E[R|\pi_A] - E[R|\pi_B] \\
E[D_N] &= E[R|\pi_A] - E[R|\pi_B] - \frac{1}{n}b_{A,B} \\
E[(D_U - E[D_U])^2] &= \frac{1}{n}(s^2_{A,A} - 2s^2_{A,B} + s^2_{B,B} - \eta^2_{A,A} + 2\eta^2_{A,B} - \eta^2_{B,B}) \\
E[(D_N - E[D_N])^2] &= \frac{1}{n}(\overline{s^2_{A,A}} - 2\overline{s^2_{A,B}} + \overline{s^2_{B,B}} - \overline{\eta^2_{A,A}} + 2\overline{\eta^2_{A,B}} - \overline{\eta^2_{B,B}}) \\
&\quad - 3\frac{1}{n}(E[R|\pi_A] - E[R|\pi_B])b_{A,B} + O(\frac{1}{n^2})\ .
\end{aligned}
$$

[2]For the normalized difference estimator, the expectations shown are for the numerator of the difference. The denominator is a positive quantity and can be scaled to be approximately 1. Because the difference is only used for comparisons, this scaling makes no difference in its performance. See the appendix for more details.

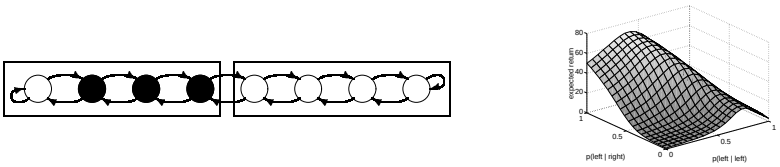


Figure 3: Left: Diagram of the "left-right" world. This world has eight states. The agent receives no reward in the outlined states and one unit of reward each time it enters one of the solid states. The agent only observes whether it is in the left or right set of boxed states (a single bit of information). Each trial begins in the fourth state from the left and lasts 100 time steps. Right: The true expected return as a function of policy for this world.

The bias of the normalized return difference estimator and the variance of both return differences estimators decreases as $\frac{1}{n}$. It is useful to note that if all of the π^i's are the same, then $\tilde{p}(h,g) = \overline{p}(h)\overline{p}(g)$ and thus $b_{A,B} = E[R|\pi_A] - E[R|\pi_B]$. In this case $E[D_N] = \frac{n-1}{n}(E[R|\pi_A] - E[R|\pi_B])$. If the estimator is only used for comparisons, this value is just as good as the true return difference (of course, for small n, the same variance would cause greater relative fluctuations).

In general we would expect $b_{A,B}$ to be of the same sign as $E[R|\pi_A] - E[R|\pi_B]$. We would also expect $\overline{s^2_{X,Y}}$ to be less than $s^2_{X,Y}$ (and similarly $\overline{\eta^2_{X,Y}}$ to be less than $\eta^2_{X,Y}$). $\overline{s^2_{X,Y}}$ and $\overline{\eta^2_{X,Y}}$ depend on the distance of the returns from the expected return under π_X and π_Y. $s^2_{X,Y}$ and $\eta^2_{X,Y}$ depend on the distance of the returns from zero. Without any other knowledge of the underlying POMDP, we fully expect that the return from an arbitrary history be closer to the expectation of the return with respect to π_X than the arbitrarily chosen value 0. If $b_{A,B}$ is the same sign as the true difference in returns and the overlined values are less than their counterparts, then the variance of the normalized estimator is less than the variance of the unnormalized estimator.

Obviously we could be unlucky and have a domain for which the unnormalized estimator has better performance. However, this seems unlikely and the reduction in variance makes up for the added bias. These intuitions are demonstrated in figure 2 where we compared the estimates for the problem described in section 4.2.

4 Experiments

4.1 Reinforcement Learning Algorithm

We can turn any of these estimators into a greedy learning algorithm. To find a policy by which to act, the agent maximizes the value of the estimator by hill-climbing in the space of policies (using the previous policy as a starting point) until it reaches a maximum. The agent uses this new policy for the next trial. After the trial, it adds the new policy-history-return triple to its data and repeats.

The hill-climbing algorithm must be carefully chosen. For many estimates, the derivative of the estimate varies greatly in magnitude (as shown below). Therefore, we have found it best to use the direction of the gradient, but not its magnitude to determine the direction in which to climb. The step size can be determined based on whether the previous step succeeded in increasing the estimated expected return. In particular, we employ a conjugate gradient descent algorithm using a golden-ratio line search[8].

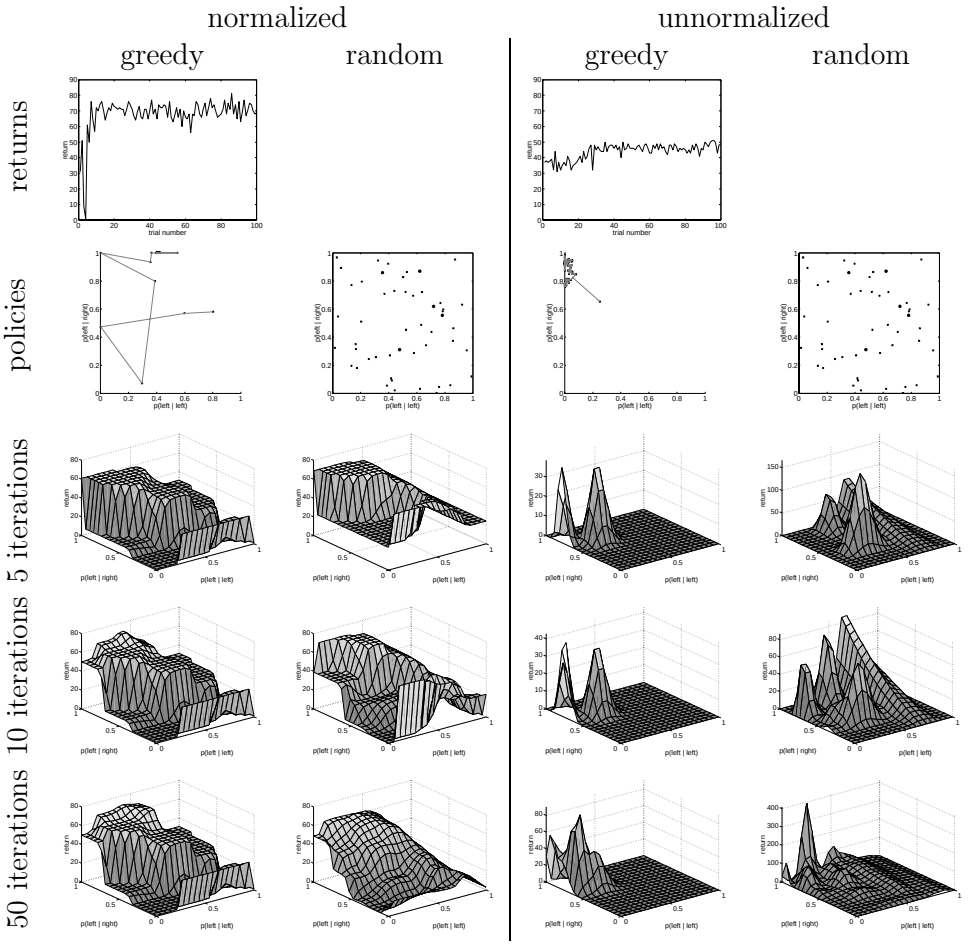


Figure 4: A comparison of the normalized and unnormalized estimators for single set of observations. For each estimator, the return estimates are shown plotted after 5, 10, and 50 iterations (samples). The left column is for the greedy policy improvement algorithm and the right column is for uniformly sampled policies. The first row shows the returns as a function of trial number. The second shows the path taken in policy space (or, for right columns, the random samples taken). Both estimators were given the same sequence of data for the random case.

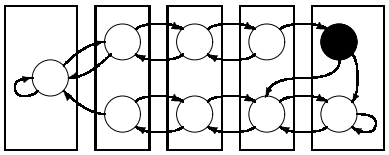
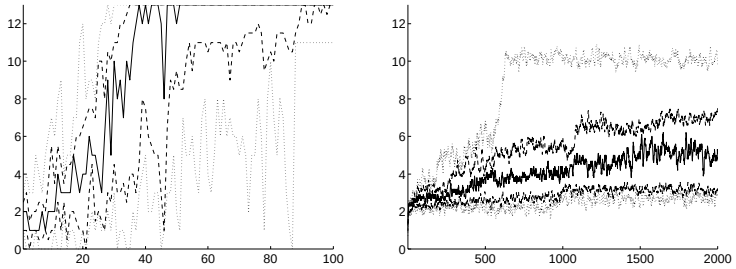


Figure 5: Diagram of the "load-unload" world. This world has nine states. The horizontal axis corresponds to the positioning of a cart. The vertical axis indicates whether the cart is loaded. The agent only observes the position of the cart (five observations denoted by boxes). The cart is loaded when it reaches the left-most state and if it reaches the right-most position while loaded, it is unloaded and the agent receives a single unit of reward. The agent has two actions at each point: move left or move right. Moving left or right off the end leaves the cart unmoved. Each trial begins in the left-most state and lasts 100 time steps.



normalized greedy algorithm REINFORCE

Figure 6: Comparison of the greedy algorithm with a normalized estimator to standard REINFORCE on the load-unload problem. Plotted are the results of ten runs of each algorithm. In the case of REINFORCE, the returns have been smoothed over ten trials. The center line is the median of the returns. The lines on either side are the first and third quartiles. The top and bottom lines are the minimum and maximum values.

4.2 Two-dimensional Problem

Figure 3 shows a simple world for which the policy can be described by two numbers (the probability of going left when in the left half and the probability of going left when in the right half) and the true expected return as a function of the policy. Figure 4 compares the normalized (equation 2) and unnormalized (equation 3) estimators both with the greedy policy improvement algorithm and under random policy choices. We feel this example is illustrative of the reasons that the normalized estimate works much better on the problems we have tried. Its bias to observed returns works well to smooth out the space. The estimator is willing to extrapolate to unseen regions where the unnormalized estimator is not. This also causes the greedy algorithm to explore new areas of the policy space whereas the unnormalized estimator gets trapped in the visited area under greedy exploration and does not successfully maximize the return function.

4.3 Twenty-dimensional Problem

Although the left-right problem was nice because the estimators could be plotted, it is very simple. The load-unload problem of figure 5 is more challenging. To achieve reasonable performance, the actions must depend on the history. We give the agent one memory bit as in section 2.6; this results in twenty independent policy parameters. REINFORCE [10] has also been used to attack a

very similar problem [6]. We compare the results of the normalized estimator with greedy search to REINFORCE in figure 6. The REINFORCE algorithm frequently gets stuck in local minima. The graph shown is for the best settings for the step size schedule of REINFORCE. In the best case, REINFORCE converges to a near optimal policy in around 500 trials (100 time steps each). The greedy algorithm run with the normalized estimate makes much better use of the data. Not only does it reuse old experience, it has an explicit model of the memory bit and therefore does not need to learn the "dynamics" of the memory. Most runs converge to the optimal policy in about 50 trials. One trial took about twice as long to converge to a slightly suboptimal policy.

5 Conclusion

We think this normalized estimator shows promise. It makes good use of the data and when combined with a greedy algorithm produces quick learning. We would like to extend it in two immediate ways. The first is to provide error estimates or bounds on the return estimate. Although we have a formula for the variance of the estimator, we still need a good estimate of this variance from the samples (the formula requires full knowledge of the POMDP). Such an estimate would allow for exploration to be incorporated into the algorithm. At the moment, the greedy algorithm only exploits the current information to achieve the best result on the next trial. If a measure of the variance of the estimator were added, the algorithm could balance exploiting the data with learning about new parts of the policy space.

Second, the estimate needs to be "sparsified." After n trials, computing the estimate (or its derivative) for a given policy takes $O(n)$ work (the inner sums can be built up as the trials progress). This makes the entire algorithm quadratic in the number of trails. However, a similar estimate could probably be achieved with fewer points. Remembering only the important trials would produce a simpler estimate.

These estimators are closely related to importance sampling. Yet, it is difficult to bring the results from importance sampling to this problem. Importance sampling usually assumes that the designer has control over the sampling distribution. In our problem, we'd like to allow the agent to use experience that was sampled in any fashion. Whereas importance sampling often asks, "given an expectation to estimate, how should I sample to reduce the variance of the estimate?" we would like to ask "given a sampling method, how best should I estimate the expectation to reduce variance?" [2] does list a number of other importance sampling methods. Unfortunately none of them are computable in this case (recall that the full probability function is not available, only one factor of it).

Finally, it may seem disturbing that we must remember which policies were used on each trail. The return doesn't really depend on the policy that the agent wants to execute; it only depends on how the agent actually does act. In theory we should be able to forget which policies were tried; doing so would allow us to use data which was not gathered with a specified policy. The policies are necessary in this paper as proxies for the unobserved state sequences. We hope in future work to remove this dependence.

References

[1] John Geweke. Bayesian inference in econometric models using monte carlo integration. *Econometrica*, 57(6):1317–1339, November 1989.

[2] Tim Hesterberg. Weighted average importance sampling and defensive mixture distributions. *Technometrics*, 37(2):185–194, 1995.

[3] Michael Kearns, Yishay Mansour, and Andrew Ng. Approximate planning in large POMDPs via reusable trajectories. In *Advances in Neural Information Processing Systems*, pages 1001–1007, 1999.

[4] T. Kloek and H. K. van Dijk. Bayesian estimates of equation system parameters: An application of integration by monte carlo. *Econometrica*, 46(1):1–19, January 1978.

[5] Nicolas Meuleau, Kee-Eung Kim, Leonid Peshkin, and Leslie Pack Kaelbling. Exploration in gradient-based reinforcement learning. unpublished, in submission, 2000.

[6] Leonid Peshkin, Nicolas Meuleau, and Leslie Pack Kaelbling. Learning policies with external memory. In *Proceedings of the Sixteenth International Conference on Machine Learning*, 1999.

[7] Doina Precup, Richard S. Sutton, and Satinder Singh. Eligibility traces for off-polcy policy evaluation. In *Proceedings of the Seventeenth International Conference on Machine Learning*, 2000.

[8] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C*. Cambridge University Press, second edition, 1992.

[9] Reuven Y. Rubinstein. *Simulation and the Monte Carlo Method*. John Wiley & Sons, 1981.

[10] Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992.

A Bias and Variance Derivations

We will assume that the policy we are evaluating is π_A and thus we are interested in estimating $E[R|\pi_A]$. Later, we will look at the difference between the estimate at point π_A and at point π_B. All sums are over the data points (1 through n) unless otherwise noted. Integrals will be over the space of histories. If the space of histories is discrete (as is the case for POMDPs), these should actually be sums. However, by using integrals for histories and sums for experience, the notation becomes clearer.

To aid in the derivation of the bias and variance of the estimators, we will define a few symbols to represent common quantities and simplify the notation:

$$\begin{aligned}
\overline{p}(h) &= \frac{1}{n}\sum_i p(h|\pi^i) \\
\tilde{p}(h,g) &= \frac{1}{n}\sum_i p(h|\pi^i)p(g|\pi^i) \\
w_A^i &= \frac{p(h^i|\pi_A)}{\overline{p}(h^i)} \\
w_B^i &= \frac{p(h^i|\pi_B)}{\overline{p}(h^i)} \\
R^i &= R(h^i) \\
r_A^i &= R^i w_A^i \\
r_B^i &= R^i w_B^i \\
R_A &= E[R|\pi_A] \\
R_B &= E[R|\pi_B]
\end{aligned}$$

not all of which will be useful immediately.

A.1 Unnormalized Estimator

First we consider the unnormalized estimator

$$U(\pi_A) = \frac{1}{n}\sum_i r_A^i \ .$$

Its mean can easily be derived as

$$\begin{aligned}
E[U(\pi_A)] &= E[\frac{1}{n}\sum_i r_A^i] \\
&= \frac{1}{n}\sum_i E[r_A^i] \\
&= \frac{1}{n}\sum_i \int R(h^i)\frac{p(h^i|\pi_A)}{\overline{p}(h^i)}p(h^i|\pi^i)\,dh^i \\
&= \frac{1}{n}\sum_i \int R(h)\frac{p(h|\pi_A)}{\overline{p}(h)}p(h|\pi^i)\,dh \\
&= \int R(h)\frac{p(h|\pi_A)}{\overline{p}(h)}\frac{1}{n}\sum_i p(h|\pi^i)\,dh \\
&= \int R(h)\frac{p(h|\pi_A)}{\overline{p}(h)}\overline{p}(h)\,dh \\
&= \int R(h)p(h|\pi_A)\,dh \\
&= E[R|\pi_A] \\
&= R_A
\end{aligned}$$

and thus $U(\pi_A)$ is an unbiased estimate of R_A. Similarly, we can derive that

$$\begin{aligned}
\frac{1}{n}\sum_i E[r_B^i] &= R_B \\
\frac{1}{n}\sum_i E[w_A^i] &= 1 \\
\frac{1}{n}\sum_i E[w_B^i] &= 1
\end{aligned}$$

which will be useful later.

We can also find the variance of this estimator:

$$\begin{aligned}
E\left[(U(\pi_A) - E[U(\pi_A)])^2\right] &= E[(\frac{1}{n}\sum_i r_A^i)^2] - R_A^2 \\
&= \frac{1}{n^2}\sum_i E[(r_A^i)^2] + \frac{1}{n^2}(\sum_i E[r_A^i])^2 - \frac{1}{n^2}\sum_i E[r_A^i]^2 - R_A^2 \\
&= \frac{1}{n}\left(\frac{1}{n}\sum_i E[(r_A^i)^2] - \frac{1}{n}\sum_i E[r_A^i]^2\right) \\
&= \frac{1}{n}\left(\int R^2(h)\frac{p(h|\pi_A)}{\overline{p}(h)}p(h|\pi_A)\,dh - \iint R(h)R(g)\frac{\tilde{p}(h,g)}{\overline{p}(h)\overline{p}(g)}p(h|\pi_A)p(g|\pi_A)\,dh\,dg\right) .
\end{aligned}$$

Although this might look a bit messy, there is sense in it. The quantity inside the parentheses is constant if, as n increases, the chosen policies (π^i) remain the same. This would happen if we kept the same set of trial policies and just kept cycling through them. In fact, if all of the π^i's are equal, then the second integral works out to just be R_A^2 as $\tilde{p}(h,g) = \overline{p}(h)\overline{p}(g)$ for this case. The first integral can be rewritten as $E[R^2\frac{p(h|\pi_A)}{\overline{p}(h)}|\pi_A]$ which looks more like a term normally associated with variances.

To make this (and future) derivations simpler, we add the following definitions

$$\begin{aligned}
s_{A,A}^2 &= \frac{1}{n}\sum_i E[(r_A^i)^2] = \int R^2(h)\frac{p(h|\pi_A)}{\overline{p}(h)}p(h|\pi_A)\,dh \\
s_{A,B}^2 = s_{B,A}^2 &= \frac{1}{n}\sum_i E[r_A^i r_B^i] = \int R^2(h)\frac{p(h|\pi_A)}{\overline{p}(h)}p(h|\pi_B)\,dh \\
s_{B,B}^2 &= \frac{1}{n}\sum_i E[(r_B^i)^2] = \int R^2(h)\frac{p(h|\pi_B)}{\overline{p}(h)}p(h|\pi_B)\,dh \\
\eta_{A,A}^2 &= \frac{1}{n}\sum_i E[r_A^i]^2 = \iint R(h)R(g)\frac{\tilde{p}(h,g)}{\overline{p}(h)\overline{p}(g)}p(h|\pi_A)p(g|\pi_A)\,dh\,dg \\
\eta_{A,B}^2 = \eta_{B,A}^2 &= \frac{1}{n}\sum_i E[r_A^i]E[r_B^i] = \iint R(h)R(g)\frac{\tilde{p}(h,g)}{\overline{p}(h)\overline{p}(g)}p(h|\pi_A)p(g|\pi_B)\,dh\,dg \\
\eta_{B,B}^2 &= \frac{1}{n}\sum_i E[r_B^i]^2 \iint R(h)R(g)\frac{\tilde{p}(h,g)}{\overline{p}(h)\overline{p}(g)}p(h|\pi_B)p(g|\pi_B)\,dh\,dg\ .
\end{aligned}$$

Thus, $\mathrm{var}(U(\pi_A)) = \frac{1}{n}(s_{A,A}^2 - \eta_{A,A}^2)$.

A.2 Unnormalized Differences

Instead of computing the mean and variance of the normalized estimator (these quantities are too convoluted to be useful), we will now look at the difference of the estimator at two different policies, π_A and π_B. If we are using the estimator to guide a greedy search, then this is closer to a quantity we care about. In fact, we really care how the maximum of the estimator compares to the true maximum. That is too difficult to calculate. However, looking at the difference gives a better sense of how the estimator works when used for comparisons.

For the unnormalized estimator, $U(\pi_A) - U(\pi_B)$ is clearly an unbiased estimate of $R_A - R_B$. The

variance of the difference can be derived, similarly to the previous derivation, to be

$$E\left[(U(\pi_A) - E[U(\pi_A)] - U(\pi_B) + E[U(\pi_B)])^2\right] = \frac{1}{n}\left(s^2_{A,A} - 2s^2_{A,B} + s^2_{B,B} - \eta^2_{A,A} + 2\eta^2_{A,B} - \eta^2_{B,B}\right) \ .$$

If we define q as

$$q_{A,B}(h) = p(h|\pi_A) - p(h|\pi_B)$$

then we can also write

$$\text{var}(U(\pi_A) - U(\pi_B)) = \frac{1}{n}\left(\int R^2(h)\frac{q^2_{A,B}(h)}{\overline{p}^2(h)}\overline{p}(h)\,dh - \iint R(h)R(g)\frac{q_{A,B}(h)q_{A,B}(g)}{\overline{p}(h)\overline{p}(g)}\tilde{p}(h,g)\,dh\,dg\right) \ .$$

A.3 Normalized Differences

For the normalized estimator we are interested in

$$\frac{\sum_i r^i_A}{\sum_i w^i_A} - \frac{\sum_i r^i_B}{\sum_i w^i_B} = \frac{\sum_i r^i_A \sum_j w^j_B - \sum_i r^i_B \sum_j w^j_A}{\sum_i w^i_A \sum_j w^i_B} \ .$$

However, since the denominator is always positive and we only care about the sign of this quantity (because we are using the estimator to compare potential policies), we can concentrate on the numerator only. Furthermore, we can scale this quantity by any positive value. We will scale it by $\frac{1}{n^2}$ so that it is roughly the same as the unnormalized estimator and the variances can be compared.

Thus, we are interested in the bias and variance of the difference

$$D = \frac{1}{n^2}\sum_{i,j}(r^i_A w^j_B - r^i_B w^j_A) \ .$$

It is important to note that $r^i_A w^i_B = r^i_B w^i_A$ and thus when $i = j$ the two terms in the sum cancel. This leaves us with

$$D = \frac{1}{n^2}\sum_{i\neq j}(r^i_A w^j_B - r^i_B w^j_A) \ .$$

The bias derivation is similar to the variance derivations of the unnormalized estimator.

$$\begin{aligned}
E[D] &= \frac{1}{n^2}\left(\sum_{i\neq j} E[r^i_A w^j_B] - \sum_{i\neq j} E[r^i_B w^j_A]\right) \\
&= \frac{1}{n^2}\left(\sum_{i,j} E[r^i_A]E[w^j_B] - \sum_i E[r^i_A]E[w^i_B] - \sum_{i,j} E[r^i_B]E[w^j_A] - \sum_i E[r^i_B]E[w^i_A]\right) \\
&= \frac{1}{n}\sum_i E[r^i_A]\frac{1}{n}\sum_j E[w^j_B] - \frac{1}{n}\sum_i E[r^i_B]\frac{1}{n}\sum_j E[w^j_A] - \frac{1}{n}\left(\frac{1}{n}\sum_i (E[r^i_A]E[w^i_B] - E[r^i_B]E[w^i_A])\right) \\
&= R_A - R_B - \frac{1}{n}b_{A,B}
\end{aligned}$$

where we define $b_{A,B}$ as

$$b_{A,B} = \iint [R(h) - R(g)]\frac{\tilde{p}(h,g)}{\overline{p}(h)\overline{p}(g)}p(h|\pi_A)p(g|\pi_B)\,dh\,dg \ .$$

Again, if we fix the relative frequencies of the selected policies, $b_{A,B}$ is a constant as n increases. Thus, the bias of D decreases at a rate of $\frac{1}{n}$. In fact, if all of the π^i's are the same, $b_{A,B} = R_A - R_B$ and the expectation of the difference is

$$\frac{n-1}{n}(R_A - R_B)$$

which, except for $n = 1$, is just a scaled version of the true difference.

The derivation of the variance of D is slightly more involved but involves the same basic technique. First, however, we must make a brief detour. Consider, in general, calculating $E[\sum_{i\neq j,k\neq l} a_i b_j c_k d_l]$. We wish to break this sum up into independent components so that the expectations can be calculated. It involves a lot of algebra but the notion is fairly simple.

First we count up the different ways that the indices could coincide.

$$\begin{aligned}\sum_{i\neq j,k\neq l} a_i b_j c_k d_l = &\sum_{i\neq j\neq k\neq l} a_i b_j c_k d_l \\ &+ \sum_{i\neq j\neq k} (a_i c_i b_j d_k + a_i d_i b_j c_k + a_i b_j c_j d_k + a_i b_j d_j c_k) \\ &+ \sum_{i\neq j} (a_i c_i b_j d_k + a_i d_i b_j c_j)\end{aligned}$$

where the notation $i \neq j \neq k \neq l$ implies none of the indices equal each other. While the expectation can now be pushed through these three sums easily (because we know the indices to be different), we have the complication that the sums are difficult to compute. We therefore break each sum into a sum over all indices minus a set of sums accounting for when the indices are the same (and therefore should not have been included). As is usual with inclusion-exclusion calculations, we also

have to add back in some sums which were subtracted out twice. The net result is,

$$
\begin{aligned}
&E[\sum_{i\neq j,k\neq l} a_i b_j c_k d_l] \\
&\quad = \sum_{i\neq j\neq k\neq l} E[a_i]E[b_j]E[c_k]E[d_l] \\
&\qquad + \sum_{i\neq j\neq k} (E[a_i c_i]E[b_j]E[d_k] + E[a_i d_i]E[b_j]E[c_k] + E[a_i]E[b_j c_j]E[d_k] + E[a_i]E[b_j d_j]E[c_k]) \\
&\qquad + \sum_{i\neq j} (E[a_i c_i]E[b_j d_j] + E[a_i d_i]E[b_j c_j]) \\
&\quad = (\sum_{i,j,k,l} E[a_i]E[b_j]E[c_k]E[d_l] \\
&\qquad - \sum_{i,j,k} E[a_i]E[b_i]E[c_j]E[d_k] + E[a_i]E[b_j]E[c_i]E[d_k] + E[a_i]E[b_j]E[c_k]E[d_i] \\
&\qquad\quad + E[a_i]E[b_j]E[c_j]E[d_k] + E[a_i]E[b_j]E[c_k]E[d_j] + E[a_i]E[b_j]E[c_k]E[d_k] \\
&\qquad + 2\sum_{i,j} E[a_i]E[b_i]E[c_i]E[d_j] + E[a_i]E[b_i]E[c_j]E[d_i] + E[a_i]E[b_j]E[c_i]E[d_i] + E[a_i]E[b_j]E[c_j]E[d_j] \\
&\qquad + \sum_{i,j} E[a_i]E[b_i]E[c_j]E[d_j] + E[a_i]E[b_j]E[c_i]E[d_j] + E[a_i]E[b_j]E[c_j]E[d_i] \\
&\qquad - 6\sum_{i} E[a_i]E[b_i]E[c_i]E[d_i]) \\
&\qquad + (\sum_{i,j,k} E[a_i c_i]E[b_j]E[d_k] - \sum_{i,j} E[a_i c_i]E[b_i]E[d_j] \\
&\qquad\quad - \sum_{i,j} E[a_i c_i]E[b_j]E[d_i] - \sum_{i,j} E[a_i c_i]E[b_j]E[d_j] + 2\sum_{i} E[a_i c_i]E[b_i]E[d_i]) \\
&\qquad + (\sum_{i,j,k} E[a_i d_i]E[b_j]E[c_k] - \sum_{i,j} E[a_i d_i]E[b_i]E[c_j] \\
&\qquad\quad - \sum_{i,j} E[a_i d_i]E[b_j]E[c_i] - \sum_{i,j} E[a_i d_i]E[b_j]E[c_j] + 2\sum_{i} E[a_i d_i]E[b_i]E[c_i]) \\
&\qquad + (\sum_{i,j,k} E[a_i]E[b_j c_j]E[d_k] - \sum_{i,j} E[a_i]E[b_i c_i]E[d_j] \\
&\qquad\quad - \sum_{i,j} E[a_i]E[b_j c_j]E[d_j] - \sum_{i,j} E[a_i]E[b_j c_j]E[d_i] + 2\sum_{i} E[a_i]E[b_i c_i]E[d_i]) \\
&\qquad + (\sum_{i,j,k} E[a_i]E[b_j d_j]E[c_k] - \sum_{i,j} E[a_i]E[b_i d_i]E[c_j] \\
&\qquad\quad - \sum_{i,j} E[a_i]E[b_j d_j]E[c_j] - \sum_{i,j} E[a_i]E[b_j d_j]E[c_i] + 2\sum_{i} E[a_i]E[b_i d_i]E[c_i]) \\
&\qquad + (\sum_{i,j} E[a_i c_i]E[b_j d_j] - \sum_{i} E[a_i c_i]E[b_i d_i]) \\
&\qquad + (\sum_{i,j} E[a_i d_i]E[b_j c_j] - \sum_{i} E[a_i d_i]E[b_i c_i])
\end{aligned}
\tag{4}
$$

and although this looks complex, most of the terms are irrelevant. For instance, consider the term $\sum_{i,j} E[a_i d_i]E[b_j c_j]$ (the very last sum). In the derivation of variance, this sum would be instantiated with $a_i = r_A^i$, $b_j = w_B^j$, $c_j = r_B^j$, and $d_i = w_A^i$. We can then rewrite it as

$$\begin{aligned}
\sum_{i,j} E[a_i d_i]E[b_j c_j] &= \sum_{i,j} E[r_A^i w_A^i]E[r_B^j w_B^j] \\
&= \sum_i \int R(h)\frac{p(h|\pi_A)p(h|\pi_A)}{\overline{p}(h)\overline{p}(h)}p(h|\pi^i)\,dh \sum_j \int R(h)\frac{p(h|\pi_B)p(h|\pi_B)}{\overline{p}(h)\overline{p}(h)}p(h|\pi^j)\,dh \\
&= n^2 \int R(h)\frac{p(h|\pi_A)p(h|\pi_A)}{\overline{p}(h)\overline{p}(h)}\frac{1}{n}\sum_i p(h|\pi^i)\,dh \int R(h)\frac{p(h|\pi_B)p(h|\pi_B)}{\overline{p}(h)\overline{p}(h)}\frac{1}{n}\sum_j p(h|\pi^j)\,dh \\
&= n^2 \int R(h)\frac{p^2(h|\pi_A)}{\overline{p}^2(h)}\overline{p}(h)\,dh \int R(h)\frac{p^2(h|\pi_B)}{\overline{p}^2(h)}\overline{p}(h)\,dh
\end{aligned}$$

where the two integrals are similar to those in equation A.1 in that they are constant quantities based on the difference between the sampling distribution and the target distributions. What is important is that converting $\sum_i p(h|\pi^i)$ into $\overline{p}(h)$ required pulling in a factor of $\frac{1}{n}$. Because there were two sums, this had to be done twice resulting in the n^2 term out in front. In general we need only to consider the highest order terms and so only the sums with three or four indices will need to be calculated. The rest will result in insignificant terms.

We can now approximate equation 4 as

$$\begin{aligned}
\sum_{i\neq j,k\neq l} E[a_i b_j c_k d_l] \approx \sum_{i,j,k,l} & E[a_i]E[b_j]E[c_k]E[d_l] \\
& - \sum_{i,j,k}\Big(E[a_i]E[b_i]E[c_j]E[d_k] + E[a_i]E[b_j]E[c_i]E[d_k] + E[a_i]E[b_j]E[c_k]E[d_i] \\
& \quad + E[a_i]E[b_j]E[c_j]E[d_k] + E[a_i]E[b_j]E[c_k]E[d_j] + E[a_i]E[b_j]E[c_k]E[d_k]\Big) \\
& + \sum_{i,j,k}\Big(E[a_i c_i]E[b_j]E[d_k] + E[a_i d_i]E[b_j]E[d_k] \\
& \quad + E[a_i]E[b_j c_j]E[d_k] + E[a_i]E[b_j d_j]E[c_k]\Big)
\end{aligned} \tag{5}$$

where the error in the approximation is $O(n^2)$.

In preparation for the variance of D, we add a few more definitions similar to those of equation A.1:

$$
\begin{aligned}
u^2_{A,A} &= \frac{1}{n}\sum_i E[r^i_A w^i_A] = \int R(h)\frac{p(h|\pi_A)}{\overline{p}(h)}p(h|\pi_A)\,dh \\
u^2_{A,B} = u^2_{B,A} &= \frac{1}{n}\sum_i E[r^i_B w^i_A] = \int R(h)\frac{p(h|\pi_A)}{\overline{p}(h)}p(h|\pi_B)\,dh \\
u^2_{B,B} &= \frac{1}{n}\sum_i E[r^i_B w^i_B] \int R(h)\frac{p(h|\pi_B)}{\overline{p}(h)}p(h|\pi_B)\,dh \\
\mu^2_{A,A} &= \frac{1}{n}\sum_i E[r^i_A]E[w^i_A] = \iint R(h)\frac{\tilde{p}(h,g)}{\overline{p}(h)\overline{p}(g)}p(h|\pi_A)p(g|\pi_A)\,dh\,dg \\
\mu^2_{A,B} &= \frac{1}{n}\sum_i E[r^i_A]E[w^i_B] = \iint R(h)\frac{\tilde{p}(h,g)}{\overline{p}(h)\overline{p}(g)}p(h|\pi_A)p(g|\pi_B)\,dh\,dg \\
\mu^2_{B,A} &= \frac{1}{n}\sum_i E[r^i_B]E[w^i_A] = \iint R(h)\frac{\tilde{p}(h,g)}{\overline{p}(h)\overline{p}(g)}p(h|\pi_B)p(g|\pi_A)\,dh\,dg \\
\mu^2_{B,B} &= \frac{1}{n}\sum_i E[r^i_B]E[w^i_B] = \iint R(h)\frac{\tilde{p}(h,g)}{\overline{p}(h)\overline{p}(g)}p(h|\pi_B)p(g|\pi_B)\,dh\,dg \qquad (6) \\
v^2_{A,A} &= \frac{1}{n}\sum_i E[(w^i_A)^2] = \int \frac{p(h|\pi_A)}{\overline{p}(h)}p(h|\pi_A)\,dh \\
v^2_{A,B} = v^2_{B,A} &= \frac{1}{n}\sum_i E[w^i_A w^i_B] = \int \frac{p(h|\pi_A)}{\overline{p}(h)}p(h|\pi_B)\,dh \\
v^2_{B,B} &= \frac{1}{n}\sum_i E[(w^i_B)^2] = \int \frac{p(h|\pi_B)}{\overline{p}(h)}p(h|\pi_B)\,dh \\
\xi^2_{A,A} &= \frac{1}{n}\sum_i E[w^i_A]^2 = \iint \frac{\tilde{p}(h,g)}{\overline{p}(h)\overline{p}(g)}p(h|\pi_A)p(g|\pi_A)\,dh\,dg \\
\xi^2_{A,B} = \xi^2_{B,A} &= \frac{1}{n}\sum_i E[w^i_A]E[w^i_B] = \iint \frac{\tilde{p}(h,g)}{\overline{p}(h)\overline{p}(g)}p(h|\pi_A)p(g|\pi_B)\,dh\,dg \\
\xi^2_{B,B} &= \frac{1}{n}\sum_i E[w^i_B]^2 = \iint \frac{\tilde{p}(h,g)}{\overline{p}(h)\overline{p}(g)}p(h|\pi_B)p(g|\pi_B)\,dh\,dg\ .
\end{aligned}
$$

We can now attack the variance of D.

$$
\begin{aligned}
E[(D-E[D])^2] &= E[D^2]-E[D]^2 \\
&= \frac{1}{n^4}E\left[\sum_{i\neq j,k\neq l}\left(r_A^i w_B^j r_A^k w_B^l - 2r_A^i w_B^j r_B^k w_A^l + r_B^i w_A^j r_B^k w_A^l\right)\right] - E[D]^2 \\
&= \left[\frac{1}{n^4}\sum_{i\neq j,k\neq l} r_A^i w_B^j r_A^k w_B^l\right] - 2\left[\frac{1}{n^4}\sum_{i\neq j,k\neq l} r_A^i w_B^j r_B^k w_A^l\right] \\
&\quad + \left[\frac{1}{n^4}\sum_{i\neq j,k\neq l} r_B^i w_A^j r_B^k w_A^l\right] - E[D]^2 \\
&= \left[R_A^2 - 4\frac{1}{n}R_A\mu_{A,B}^2 - \frac{1}{n}\eta_{A,A}^2 - \frac{1}{n}R_A^2\xi_{B,B}^2 + 4\frac{1}{n}R_A u_{A,B}^2 + \frac{1}{n}s_{A,A}^2 + \frac{1}{n}R_A^2 v_{B,B}^2\right] \\
&\quad - 2\Bigg[R_A R_B \\
&\qquad - \frac{1}{n}R_A\mu_{B,A}^2 - \frac{1}{n}R_A\mu_{B,B}^2 - \frac{1}{n}R_B\mu_{A,B}^2 - \frac{1}{n}R_B\mu_{A,A}^2 - \frac{1}{n}\eta_{A,B}^2 - \frac{1}{n}R_A R_B\xi_{A,B}^2 \\
&\qquad + \frac{1}{n}R_A u_{A,B}^2 + \frac{1}{n}R_A u_{B,B}^2 + \frac{1}{n}R_B u_{A,B}^2 + \frac{1}{n}R_B u_{A,A}^2 + \frac{1}{n}s_{A,B}^2 + \frac{1}{n}R_A R_B v_{A,B}^2\Bigg] \\
&\quad + \left[R_B^2 - 4\frac{1}{n}R_B\mu_{B,A}^2 - \frac{1}{n}\eta_{B,B}^2 - \frac{1}{n}R_B^2\xi_{A,A}^2 + 4\frac{1}{n}R_B u_{A,B}^2 + \frac{1}{n}s_{B,B}^2 + \frac{1}{n}R_B^2 v_{A,A}^2\right] \\
&\quad - R_A^2 - R_B^2 + 2R_A R_B + \frac{1}{n}(R_A - R_B)b_{A,B} + O(\frac{1}{n^2}) \\
&= \frac{1}{n}\Bigg[(R_A^2 v_{B,B}^2 - 2R_A u_{B,B}^2 + s_{B,B}^2) - (R_A^2\xi_{B,B}^2 - 2R_A\mu_{B,B}^2 + \eta_{B,B}^2) \\
&\qquad (R_B^2 v_{A,A}^2 - 2R_B u_{A,A}^2 + s_{A,A}^2) - (R_B^2\xi_{A,A}^2 - 2R_B\mu_{A,A}^2 + \eta_{A,A}^2) \\
&\qquad - 2(R_A R_B v_{A,B}^2 - R_A u_{A,B}^2 - R_B u_{A,B}^2 + s_{A,B}^2) \\
&\qquad + 2(R_A R_B\xi_{A,B}^2 - R_A\mu_{B,A}^2 - R_B\mu_{A,B}^2 + \eta_{A,B}^2) \\
&\qquad - 4(R_A - R_B)(\mu_{A,B}^2 - \mu_{B,A}^2) + (R_A - R_B)b_{A,B}\Bigg] + O(\frac{1}{n^2})\ .
\end{aligned}
\tag{7}
$$

The fourth line came from applying the expansion of equation 5 along with the definitions from equations A.1 and 6.

At this point we need to introduce a few new definitions for the final bit of algebra.

$$
\begin{aligned}
\overline{s^2_{A,A}} &= \frac{1}{n}\sum_i E[(r^i_A)^2] = \int (R(h)-R_A)^2 \frac{p(h|\pi_A)}{\overline{p}(h)} p(h|\pi_A)\, dh \\
\overline{s^2_{A,B}} = \overline{s^2_{B,A}} &= \frac{1}{n}\sum_i E[r^i_A r^i_B] = \int (R(h)-R_A)(R(h)-R_B) \frac{p(h|\pi_A)}{\overline{p}(h)} p(h|\pi_B)\, dh \\
\overline{s^2_{B,B}} &= \frac{1}{n}\sum_i E[(r^i_B)^2] = \int (R(h)-R_B)^2 \frac{p(h|\pi_B)}{\overline{p}(h)} p(h|\pi_B)\, dh \\
\overline{\eta^2_{A,A}} &= \frac{1}{n}\sum_i E[r^i_A]^2 = \iint (R(h)-R_A)(R(g)-R_A) \frac{\tilde{p}(h,g)}{\overline{p}(h)\overline{p}(g)} p(h|\pi_A) p(g|\pi_A)\, dh\, dg \\
\overline{\eta^2_{A,B}} = \overline{\eta^2_{B,A}} &= \frac{1}{n}\sum_i E[r^i_A] E[r^i_B] = \iint (R(h)-R_A)(R(g)-R_B) \frac{\tilde{p}(h,g)}{\overline{p}(h)\overline{p}(g)} p(h|\pi_A) p(g|\pi_B)\, dh\, dg \\
\overline{\eta^2_{B,B}} &= \frac{1}{n}\sum_i E[r^i_B]^2 \iint (R(h)-R_B)(R(g)-R_B) \frac{\tilde{p}(h,g)}{\overline{p}(h)\overline{p}(g)} p(h|\pi_B) p(g|\pi_B)\, dh\, dg \ .
\end{aligned}
\tag{8}
$$

Most usefully, the following identities hold.

$$
\begin{aligned}
\overline{s^2_{A,A}} &= R_A^2 v^2_{A,A} - 2R_A u^2_{A,A} - s^2_{A,A} \\
\overline{s^2_{A,B}} &= R_A R_B v^2_{A,B} - R_A u^2_{A,B} - R_B u^2_{A,B} + s^2_{A,B} \\
\overline{s^2_{B,B}} &= R_B^2 v^2_{B,B} - 2R_B u^2_{B,B} + s^2_{B,B} \\
\overline{\eta^2_{A,A}} &= R_A^2 \xi^2_{A,A} - 2R_A \mu^2_{A,A} + \eta^2_{A,A} \\
\overline{\eta^2_{A,B}} &= R_A R_B \xi^2_{A,B} - R_A \mu^2_{B,A} - R_B \mu^2_{A,B} + \eta^2_{A,B} \\
\overline{\eta^2_{B,B}} &= R_B^2 \xi^2_{B,B} - 2R_B \mu^2_{B,B} + \eta^2_{B,B} \\
b_{A,B} &= \mu^2_{A,B} - \mu^2_{B,A} \ .
\end{aligned}
$$

The variance of D can now be expressed as (continuing from equation 7)

$$
\begin{aligned}
E[(D-E[D])^2] = \frac{1}{n}\Big(&\overline{s^2_{A,A}} - 2\overline{s^2_{A,B}} + \overline{s^2_{B,B}} - \overline{\eta^2_{A,A}} + 2\overline{\eta^2_{A,B}} - \overline{\eta^2_{B,B}}\Big) \\
&- 3\frac{1}{n}(R_A - R_B) b_{A,B} + O(frac1n^2) \ .
\end{aligned}
$$

The first line is better than the variance of the unnormalized difference. The equation is the same except each quantity has been replaced by the "overlined" version. If we compare the two versions (equations A.1 and 8) we can note that the non-overlined versions are all integrals averaging the $R(h)$ by some positive weights. However for the overlined versions, R_A or R_B are subtracted from the returns before the averaging. We expect that R_A and R_B to be closer to the mean of these quantities than 0 and thus the overlined quantities are smaller. In general, the variance of the normalized estimator is invariant to translation of the returns (which is not surprising given that the estimator is invariant in the same way). The unnormalized estimator is not invariant in this manner. If the sampled π^i's are all the same, $b_{A,B} = R_A - R_B$ and thus the second line is a quantity less than zero plus a term that is only order $\frac{1}{n^2}$.