Weighted importance sampling for off-policy learning with linear function approximation

Authored by:

Richard S. Sutton
A. Rupam Mahmood
Hado P. van Hasselt

Abstract

Importance sampling is an essential component of off-policy model-free reinforcement learning algorithms. However, its most effective variant, emph{weighted} importance sampling, does not carry over easily to function approximation and, because of this, it is not utilized in existing off-policy learning algorithms. In this paper, we take two steps toward bridging this gap. First, we show that weighted importance sampling can be viewed as a special case of weighting the error of individual training samples, and that this weighting has theoretical and empirical benefits similar to those of weighted importance sampling. Second, we show that these benefits extend to a new weighted-importance-sampling version of off-policy LSTD(lambda). We show empirically that our new WIS-LSTD(lambda) algorithm can result in much more rapid and reliable convergence than conventional off-policy LSTD(lambda) (Yu 2010, Bertsekas & Yu 2009).

1 Paper Body

Importance sampling is an essential component of off-policy model-free reinforcement learning algorithms. However, its most effective variant, weighted importance sampling, does not carry over easily to function approximation and, because of this, it is not utilized in existing off-policy learning algorithms. In this paper, we take two steps toward bridging this gap. First, we show that weighted importance sampling can be viewed as a special case of weighting the error of individual training samples, and that this weighting has theoretical and empirical benefits similar to those of weighted importance sampling. Second, we show that these benefits extend to a new weighted-importance-sampling version of offpolicy LSTD( ). We show empirically that our new WIS-LSTD( ) algorithm can result in much more rapid and reliable convergence than conventional off-policy LSTD( ) (Yu 2010, Bertsekas & Yu 2009).

1

Importance sampling and weighted importance sampling

Importance sampling (Kahn & Marshall 1953, Rubinstein 1981, Koller & Friedman 2009) is a wellknown Monte Carlo technique for estimating an expectation under one distribution given samples from a different distribution. Consider that data samples Yk 2 R are generated i.i.d. from a sample . distribution l, but we are interested in estimating the expected value of these samples, vg = E g [Yk], under a different distribution g. In importance sampling this is achieved simply by averaging the . k) samples weighted by the ratio of their likelihoods ?k = g(Y l(Yk) , called the importance-sampling ratio. That is, vg is estimated as: Pn ?k Y k . v?g = k=1 . (1) n This is an unbiased estimate because each of the samples it averages is unbiased: Z Z g(y) E l [?k Yk] = l(y) y dy = g(y)y dy = E g [Yk] = vg . l(y) y y Unfortunately, this importance sampling estimate is often of unnecessarily high variance. To see how this can happen, consider a case in which the samples Yk are all nearly the same (under both distributions) but the importance-sampling ratios ?k vary greatly from sample to sample. This should be an easy case because the samples are so similar for the two distributions, but importance sampling will average the ?k Yk , which will be of high variance, and thus its estimates will also be of high variance. In fact, without further bounds on the importance-sampling ratios, v?g may have infinite variance (Andrad?ottir et al. 1995, Robert & Casella 2004).

An important variation on importance sampling that often has much lower variance is weighted importance sampling (Rubinstein 1981, Koller & Friedman 2009). The weighted importance sampling 1

(WIS) estimate vg as a weighted average of the samples with importance-sampling ratios as weights: Pn . k=1 ?k Yk v?g = P . n k=1 ?k

This estimate is biased, but consistent (asymptotically correct) and typically of much lower variance than the ordinary importance-sampling (OIS) estimate, as acknowledged by many authors (Hesterberg 1988, Casella & Robert 1998, Precup, Sutton & Singh 2000, Shelton 2001, Liu 2001, Koller & Friedman 2009). For example, in the problematic case sketched above (near constant Yk , widely varying ?k) the variance of the WIS estimate will be related to the variance of Yk . Note also that when the samples are bounded, the WIS estimate has bounded variance, because the estimate itself is bounded by the highest absolute value of Yk , no matter how large the ratios ?k are (Precup, Sutton & Dasgupta 2001). Although WIS is the more successful importance sampling technique, it has not yet been extended to parametric function approximation. This is problematic for applications to off-policy reinforcement learning, in which function approximation is viewed as essential for large-scale applications to sequential decision problems with large state and action spaces. Here an important subproblem is the approximation of the value function?the expected sum of future discounted rewards as a function of state?for a designated target policy that may differ from that used to select actions. The existing methods for off-policy value-function approximation either use OIS (Maei & Sutton 2010, Yu 2010, Sutton et al. 2014, Geist & Scherrer 2014, Dann et al. 2014) or use WIS but are limited to the tabular or non-parametric case (Precup et al. 2000, Shelton

2001). How to extend WIS to parametric function approximation is important, but far from clear (as noted by Precup et al. 2001).

2

Importance sampling for linear function approximation

In this section, we take the first step toward bridging the gap between WIS and off-policy learning with function approximation. In a general supervised learning setting with linear function approximation, we develop and analyze two importance-sampling methods. Then we show that these two methods have theoretical properties similar to those of OIS and WIS. In the fully-representable case, one of the methods becomes equivalent to the OIS estimate and the other to the WIS estimate. The key idea is that OIS and WIS can be seen as least-squares solutions to two different empirical objectives. The OIS estimate is the least-squares solution to an empirical mean-squared objective where the samples are importance weighted: Pn n n X ?k Y k 1X 2 v?g = arg min (?k Yk v) =) (?k Yk v?g) = 0 =) v?g = k=1 . (2) n n v k=1

k=1

Similarly, the WIS estimate is the least-squares solution to an empirical mean-squared objective where the individual errors are importance weighted: Pn n n X 1X 2 k=1 ?k Yk v?g = arg min ?k (Yk v) =) ?k (Yk v?g) = 0 =) v?g = P . (3) n n v k=1 ?k k=1

k=1

We solve similar empirical objectives in a general supervised learning setting with linear function approximation to derive the two new methods. Consider two correlated random variables Xk and Yk , where Xk takes values from a finite set X , and where Yk 2 R. We want to estimate the conditional expectation of Yk for each x 2 X under a target distribution gY —X . However, the samples (Xk , Yk) are generated i.i.d. according to a joint sample distribution lXY (?) with conditional probabilities lY —X that may differ from the conditional . target distribution. Each input is mapped to a feature vector k = (Xk) 2 Rm , and the goal is to estimate the expectation E gY —X [Yk — Xk = x] as a linear function of the features . ? ¿ (x) ? vg (x) = E gY —X [Yk —Xk = x] .

Estimating this expectation is again difficult because the target joint distribution of the input-output pairs gXY can be different than the sample joint distribution lXY . Generally, the discrepancy in 2

the joint distribution may arise from two sources: difference in marginal distribution of inputs, gX 6= lX , and difference in the conditional distribution of outputs, gY —X 6= lY —X . Problems where only the former discrepancy arise are known as covariate shift problems (Shimodaira 2000). In these problems the conditional expectation of the outputs is assumed unchanged between the target and the sample distributions. In off-policy learning problems, the discrepancy between conditional probabilities is more important. Most off-policy learning methods correct only the discrepancy between the target and the sample conditional distributions of outputs (Hachiya et al. 2009, Maei & Sutton 2010, Yu 2010, Maei 2011, Geist & Scherrer 2014, Dann et al. 2014). In this paper, we also focus only on correcting the discrepancy between the conditional distributions. The problem of estimating vg (x) as a linear function of features

using samples generated from l can be formulated as the minimization of the mean squared error (MSE) where the solution is as follows: ?? ?2 ? ? 1 ? ? ?? = ? arg min E lX E gY —X [Yk —Xk] ? ¿ k = E lX k ¿ E lX E gY —X [Yk —Xk] k . (4) k ?

Similar to the empirical mean-squared objectives defined in (2) and (3), two different empirical objectives can be defined to approximate this solution. In one case the importance weighting is applied to the output samples, Yk , and in the other case the importance weighting is applied to the error, Yk ? ¿ k , n ? n ?2 ? ?2 . 1X . 1X J?n (?) = ?k Y k ? ¿ k ; J?n (?) = ?k Y k ? ¿ k , n n k=1

k=1

. where importance-sampling ratios are defined by ?k = gY —X (Yk —Xk)/lY —X (Yk —Xk).

We can minimize these objectives by equating the derivatives of the above empirical objectives to zero. Provided the relevant matrix inverses exist, the resulting solutions are, respectively ! 1 n n X X . ¿ ?n = ? ?k Yk k , and (5) k k

k=1

. ?n = ?

n X

k=1

?k

k

¿ k

!

k=1 1 n

X

?k Y k

k

.

(6)

k=1

? the OIS-LS estimator and ? ? the WIS-LS estimator. We call ? A least-squares method similar to WIS-LS above was introduced for covariate shift problems by Hachiya, Sugiyama and Ueda (2012). Although superficially similar, that method uses importancesampling ratios to correct for the discrepancy in the marginal distributions of inputs, whereas WIS-LS corrects for the discrepancy in the conditional expectations of the outputs. For the fullyrepresentable case, unlike WIS-LS, the method of Hachiya et al. becomes an ordinary Monte Carlo estimator with no importance sampling.

3

Analysis of the least-squares importance-sampling methods

The two least-squares importance-sampling methods have properties similar to those of the OIS and the WIS methods. In Theorems 1 and 2, we prove that when vg can be represented as a linear function of the features, then OIS-LS is an unbiased estimator of ? ? , whereas WIS-LS is a biased estimator, similar

to the WIS estimator. If the solution is not linearly representable, least-squares methods are not generally unbiased. In Theorem 3 and 4, we show that both least-squares estimators are consistent for ? ? . Finally, we demonstrate that the least-squares methods are generalizations of OIS and WIS by showing, in Theorem 5 and 6, that in the fully representable case (when the features form an orthonormal basis) OIS-LS is equivalent to OIS and WIS-LS is equivalent to WIS. Theorem 1. If vg is a linear function of the features, that is, vg (x) = ? ¿ ? (x), then OIS-LS is an ? n] = ?? . unbiased estimator, that is, E lXY [? Theorem 2. Even if vg is a linear function of the features, that is, vg (x) = ? ¿ ? (x), WIS-LS is in ? n] 6= ? ? . general a biased estimator, that is, E lXY [? 3

? n is a consistent estimator of the MSE solution ? ? given in (4). Theorem 3. The OIS-LS estimator ? ? n is a consistent estimator of the MSE solution ? ? given in (4). Theorem 4. The WIS-LS estimator ?

? ¿ (x) of input Theorem 5. If the features form an orthonormal basis, then the OIS-LS estimate ? n x is equivalent to the OIS estimate of the outputs corresponding to x. ? ¿ (x) of input Theorem 6. If the features form an orthonormal basis, then the WIS-LS estimate ? n x is equivalent to the WIS estimate of the outputs corresponding to x. Proofs of Theorem 1-6 are given in the Appendix. The WIS-LS estimate is perhaps the most interesting of the two least-squares estimates, because it generalizes WIS to parametric function approximation for the first time and extends its advantages.

4

A new off-policy LSTD( ) with WIS

In sequential decision problems, off-policy learning methods based on important sampling can suffer from the same high-variance issues as discussed above for the supervised case. To address this, we extend the idea of WIS-LS to off-policy reinforcement learning and construct a new off-policy WISLSTD( ) algorithm. We first explain the problem setting. Consider a learning agent that interacts with an environment where at each step t the state of the environment is St and the agent observes a feature vector . m t = (St) 2 R . The agent takes an action At based on a behavior policy b(?—St), that is typically a function of the state features. The environment provides the agent a scalar (reward) signal Rt+1 and transitions to state St+1 . This process continues, generating a trajectory of states, actions and rewards. The goal is to estimate the values of the states under the target policy ?, defined as the expected returns given by the sum of future discounted rewards: ”1 # t X Y . v? (s) = E Rt+1 (Sk) — S0 = s, At ? ?(?—St), 8t , t=0

k=1

where (Sk) 2 [0, 1] is a state-dependent degree of discounting on arrival in Sk (as in Sutton et al. 2014). We assume the rewards and discounting are chosen such that v? (s) is well-defined and finite. Our primary objective is to estimate v? as a linear function of the features: v? (s) ? ? ¿ (s), where ? 2 Rm is a parameter vector to be estimated. As before, we need to correct for the difference in sample distribution resulting from the behavior policy and the target distribution as induced by the target policy. Consider a partial trajectory

from time step k to time t, consisting of a sequence Sk , Ak , Rk , Sk+1 , . . . , St . The probability of this trajectory occurring given it starts at Sk under the target policy will generally differ from its probability under the behavior policy. The importancesampling ratio ?tk is defined to be the ratio of these probabilities. This importance-sampling ratio can be written in terms of the product of action-selection probabilities without needing a model of the environment (Sutton & Barto 1998): Qt 1 t 1 t 1 ?(Ai —Si) Y ?(Ai —Si) Y t . ?k = Qi=k = = ?i , t 1 b(Ai —Si) i=k b(Ai —Si) i=k

i=k

. where we use the shorthand ?i = ?i+1 = ?(Ai —Si)/b(Ai —Si). i

We incorporate a common technique to reinforcement learning (RL) where updates are estimated by bootstrapping, fully or partially, on previously constructed state-value estimates. Bootstrapping potentially reduces the variance of the updates compared to using full returns and makes RL algorithms applicable to non-episodic tasks. In this paper we assume that the bootstrapping parameter (s) 2 [0, 1] may depend on the state s (as in Sutton & Barto 1998, Maei & Sutton 2010). In the . . following, we use the notational shorthands k = (Sk) and k = (Sk). Following Sutton et al. (2014), we construct an empirical loss as a sum of pairs of squared corrected and uncorrected errors, each corresponding to a different number of steps of lookahead, and each . weighted as a function of the intervening discounting and bootstrapping. Let Gtk = Rk+1 + . . . + Rt be the undiscounted return truncated after looking ahead t k steps. Imagine constructing the 4

empirical loss for time 0. After leaving S0 and observing R1 and S1 , the first uncorrected error is G10 ? ¿ 0 , with weight equal to the probability of terminating 1 1 . If we do not terminate, then we continue to S1 and form the first corrected error G10 + v ¿ 1 ? ¿ 0 using the bootstrapping estimate v ¿ 1 . The weight on this error is 1 (1 1), and the parameter vector v may differ from ?. Continuing to the next time step, we obtain the second uncorrected error G20 ? ¿ 0 and the second corrected error G20 +v ¿ 2 ? ¿ 0 , with respective weights 1 1 (1 2) and 1 1 2 (1 2). This goes on until we reach the horizon of our data, say at time t, when we bootstrap fully with v ¿ t , generating a final corrected return error of Gt0 + v ¿ t ? ¿ 0 with weight 1 1 ? ? ? t 1 t 1 t . . The general form for the uncorrected error is ?tk (?) = Gtk ? ¿ k , and the general form for the . corrected error is ?kt (?, v) = Gtk + v ¿ t ? ¿ k . All these errors could be squared, weighted by their weights, and summed to form the overall empirical loss. In the off-policy case, we need to also weight the squares of the errors ?tk and ?kt by the importance-sampling ratio ?tk . Hence, the overall empirical loss at time k for data up to time t can be written as ‘tk (?, v)

t 1 X . = ?k Cki

1

i=k+1

+

?k Ckt 1

?

(1

h
i)
(1
?
t)
?ik (?)
?2
2 ?tk (?)
+ +
i (1
i)
?
?ki (?, v)
?kt (?, v) 2
t
i
?2
t . Y , where Ckt =
j
j ?j .
j=k+1

This loss differs from that used by other LSTD( ) methods in that importance weighting is applied to the individual errors within ‘tk (?, v). Now, we are ready to state the least-squares problem. As noted by Geist & Scherrer (2014), LSTD( ) methods can be derived by solving least-squares problems where estimates at each step are matched with multi-step returns starting from those steps given that bootstrapping is done using the solution itself. Our proposed new method, called WIS-LSTD( ), computes at each time t the solution to the least-squares problem: t 1 X . ? t = arg min ‘tk (?, ? t). ?

k=0

At the solution, the derivative of the objective is zero: Pt 1 ? ? k=0 2 k,t (? t , ? t) k = 0, where the errors k,t are defined by ? k,t (?, v)

t 1 X . = ?k Cki i=k+1
1
?

Next, we separate the terms of ? k,t (? t , ? t)

k
1
+
?i i) k (?, v)
i (1
+ ?k Ckt ? k,t (? t , ? t)
k
?
1
?

Pt
1 t k=0 ‘k (?, ? t) ?=? t
t t)?k (?)
(1
?t t k (?, v)
+
that involve ? t from those that do not:
=
?
.
Ak,t ? t , where bk,t 2 Rm , Ak,t 2 Rm?m and they are defined as
= bk,t
t 1 X . bk,t = ?k Cki
i i)?k (?)
(1
@ @?
(1
i i i)Gk
+ ?k Ckt
k
1
Gtk
k,
i=k+1
t 1 X . Ak,t = ?k Cki
1
k ((1
i i)
k
i (1
i)
i)
¿
+ ?k Ckt
1
k(
k
t
t)
¿
.
i=k+1
Therefore, the solution can be found as follows: t 1 X
k=0
t 1
(bk,t

. X Ak,t ? t) = 0 =) ? t = At 1 bt , where At = Ak,t , k=0
t 1
. X bt = bk,t .
(7)
k=0

In the following we show that WIS-LS is a special case of the above algorithm defined by (7). As Theorem 6 shows that WIS-LS generalizes WIS, it follows that the above algorithm generalizes WIS as well. Theorem 7. At termination, the algorithm defined by (7) is equivalent to the WIS-LS method in the ? t as sense that if 0 = ? ? ? = t = 0 = ? ? ? = t 1 = 1 and t = 0, then ? t defined in (7) equals ? . defined in (6), with Yk = Gtk . (Proved in the Appendix). 5

Our last challenge is to find an equivalent efficient online algorithm for this method. The solution in (7) cannot be computed incrementally in this form. When a new sample arrives at time t + 1, Ak,t+1 and bk,t+1 have to be computed for each k = 0, . . . , t, and hence the computational complexity of this solution grows with time. It would be preferable if the solution at time t + 1 could be computed incrementally based on the estimates from time t, requiring only constant computational complexity per time step. It is not immediately obvious such an efficient update exists. For instance, for = 1 this method achieves full Monte Carlo (weighted) importance-sampling estimation, which means whenever the target policy deviates from the behavior policy all previously made updates have to be unmade so that no updates are made towards a trajectory which is impossible under the target policy. Sutton et al. (2014) show it is possible to derive efficient updates in some cases with the use of provisional parameters which keep track of the provisional updates that might need to be unmade when a deviation occurs. In the following, we show that using such provisional parameters it is also possible to achieve an equivalent efficient update for (7). We first write both bk,t and Ak,t recursively in t (derivations in Appendix A.8): bk,t+1 = bk,t + ?k Ckt Rt+1
k
+ (?t
t 1 t t t ?k Ck Gk
1)
k,
¿
¿ Ak,t+1 = Ak,t + 1) t t ?k Ckt 1 k (k k(t t+1 t+1) + (?t t) . Using the above recursions, we can write the updates of both bt and At incrementally. The vector bt can be updated incrementally as t t 1 t 1 t 1 X X X X bt+1 = bk,t+1 = bk,t+1 + bt,t+1 = bk,t + ?t Rt+1 t + Rt+1 ?k Ckt k
?k Ckt
k=0
k=0
+ (?t
1)
t t
k=1

t 1 X
?k Ckt
1
Gtk
k=1
= bt + Rt+1 et + (?t
k
(8)
1)ut ,
k=1

where the eligibility trace et 2 Rm and the provisional vector ut 2 Rm are defined as follows: e t = ?t
+
t
t 1 X
?k Ckt
= ?t
k
t
+ ?t
?t
t t
1
t 1
k=1
ut =
t t
t 1 X
?k Ckt 1
?k Ckt t 2 X
1
Gtk
k
=
?t
t t
1 t 1 t 1
?k Ckt 1
k
+ ?t
1 Rt
t 1
k=1
t 2 X
?k Ckt
2

k=0
+
1
(?t
1 ut 1
k(
!
=
t
t+1
¿ t+1) + (?t
1)
t t
t(
t t
t 1 X
t
t+1)
t+1
+ (?t
1)Vt ,
1 t 1
k=1
+
t t
?k Ckt 1 ?t
(9)
k
+ Rt e t
t
?k Ckt
t+1)
t+1
(10)
1) .
¿
1
k(
t)
k
¿
k(
t 1
t)
¿
+ ?t

t 1(
1
t 1
1 Vt 1
+ et
1(
t 1
t)
¿
(11)
t 2 X
?k Ckt
2
k(
k
t 1)
¿
k=1 t)
¿
k=1
=
t t et 1),
k=1 ¿
where the provisional matrix Vt 2 Rm?m is defined as t 1 X ¿ Vt = t t ?k Ckt 1 k (k t) = t t ?t 1 t t 2 X
+
t
k=0
k=1
= At + et (
= ?t (
k=1
k=0
?k Ckt
!
Gtk
The matrix At can be updated incrementally as t t 1 t 1 X X X At+1 = Ak,t+1 = Ak,t+1 + At,t+1 = Ak,t + ?t t 1 X
k
k=1
k=1
+ Rt
+
t 2 X
! (12)
.

Then the parameter vector can be updated as: ? t+1 = (At+1)
1
bt+1 .
(13)

Equations (8?13) comprise our WIS-LSTD( ). Its per-step computational complexity is O(m3), where m is the number of features. The computational cost of this method does not increase with time. At present we are unsure whether or not there is an O(m2) implementation. 6

Theorem 8. The off-policy LSTD( ) method defined in (8?13) is equivalent to the off-policy LSTD( ) method defined in (7) in the sense that they compute the same ? t at each time t. Proof. The result follows immediately from the above derivation. It is easy to see that in the on-policy case this method becomes equivalent to on-policy LSTD( ) (Boyan 1999) by noting that the third term of both bt and At updates in (8) and (11) becomes zero, because in the on-policy case all the importance-sampling ratios are 1. Recently Dann et al. (2014) proposed another least-squares based off-policy method called recursive LSTD-TO( ). Unlike our algorithm, that algorithm does not specialize to WIS in the fully representable case, and it does not seem as closely related to WIS. The Adaptive Per-Decision Importance Weighting (APDIW) method by Hachiya et al. (2009) is superficially similar to WIS-LSTD( ), there are several important differences. APDIW is a one-step method that always fully bootstraps whereas WIS-LSTD( ) covers the full spectrum of multi-step backups including both one-step backup and Monte Carlo update. In the fully representable case, APDIW does not become equivalent to the WIS estimate, whereas WIS-LSTD(1) does. Moreover, APDIW does not find a consistent estimation of the off-policy target whereas WIS algorithms do.

5

Experimental results

We compared the performance of the proposed WIS-LSTD( ) method with the conventional offpolicy LSTD( ) by Yu (2010) on two random-walk tasks for off-policy policy evaluation. These random-walk tasks consist of a Markov chain with 11 non-terminal and two terminal states. They can be imagined to be laid out horizontally, where the two terminal states are at the left and the right ends of the chain. From each non-terminal state, there are two actions available: left, which leads to the state to the left and right, which leads to the state to the right. The reward is 0 for all transitions except for the rightmost transition to the terminal state, where it is +1. The initial state was set to the state in the middle of the chain. The behavior policy chooses an action uniformly randomly, whereas the target policy chooses the right action with probability 0.99. The termination function was set to 1 for the non-terminal states and 0 for the terminal states. We used two tasks based on this Markov chain in our experiments. These tasks differ by how the non-terminal states were mapped to features. The terminal states were always mapped to a vector with all zero elements. For each non-terminal state, the features were normalized so that the L2 norm of each feature vector was one. For the first task, the feature representation was tabular, that is, the feature vectors were standard

basis vectors. In this representation, each feature corresponded to only one state. For the second task, the feature vectors were binary representations of state indices. There were 11 non-terminal states, hence each feature vector had blog2 (11)c + 1 = 4 components. These vectors for the states from left to right were (0, 0, 0, 1)¿ , (0, 0, 1, 0)¿ , (0, 0, 1, 1)¿ , . . . , (1, 0, 1, 1)¿ , which were then normalized to get unit vectors. These features heavily underrepresented the states, due to the fact that 11 states were represented by only 4 features. We tested both algorithms for different values of constant , from 0 to 0.9 in steps of 0.1 and from 0.9 to 1.0 in steps of 0.025. The matrix to be inverted in both methods was initialized to ?I, where the regularization parameter ? was varied by powers of 10 with powers chosen from -3 to +3 in steps of 0.2. Performance was measured as the empirical mean squared error (MSE) between the estimated value of the initial state and its true value under the target policy projected to the space spanned by the given features. This error was measured at the end of each of 200 episodes for 100 independent runs. Figure 1 shows the results for the two tasks in terms of empirical convergence rate, optimum performance and parameter sensitivity. Each curve shows MSE together with standard errors. The first row shows results for the tabular task and the second row shows results for the function approximation task. The first column shows learning curves using (, ?) = (0, 1) for the first task and (0.95, 10) for the second. It shows that in both cases WIS-LSTD( ) learned faster and gave lower error throughout the period of learning. The second column shows performance with respect to different optimized over ?. The x-axis is plotted in a reverse log scale, where higher values are more spread out than the lower values. In both tasks, WIS-LSTD( ) outperformed the conventional LSTD( ) for all values of . For the best parameter setting (best and ?), WIS-LSTD( ) outperformed LSTD( ) by an order 7

Tabular task o?-policy LSTD( ) MSE
MSE
MSE ? 0.0 ... 0.5 ?? 0.9
WIS-LSTD( )
episodes
regularization parameter ?
Func. approx. task
o?-policy LSTD( ) MSE
MSE
MSE ? 0.5 ... 0.9 ?? 1.0
WIS-LSTD( )
episodes
regularization parameter ?

Figure 1: Empirical comparison of our WIS-LSTD( ) with conventional off-policy LSTD( ) on two random-walk tasks. The empirical Mean Squared Error shown is for the initial state at the end of each episode, averaged over 100 independent runs (and also over 200 episodes in column 2 and 3). of magnitude. The third column shows performance with respect to the regularization parameter ? for three representative values of . For a wide range of ?, WIS-LSTD( )

outperformed conventional LSTD( ) by an order of magnitude. Both methods performed similarly for large ?, as such large values essentially prevent learning for a long period of time. In the function approximation task when smaller values of ? were chosen, close to 1 led to more stable estimates, whereas smaller introduced high variance for both methods. In both tasks, the better-performing regions of ? (the U-shaped depressions) were wider for WIS-LSTD( ).

6

Conclusion

Although importance sampling is essential to off-policy learning and has become a key part of modern reinforcement learning algorithms, its most effective form?WIS?has been neglected because of the difficulty of combining it with parametric function approximation. In this paper, we have begun to overcome these difficulties. First, we have shown that the WIS estimate can be viewed as the solution to an empirical objective where the squared errors of individual samples are weighted by the importance-sampling ratios. Second, we have introduced a new method for general supervised learning called WIS-LS by extending the error-weighted empirical objective to linear function approximation and shown that the new method has similar properties as those of the WIS estimate. Finally, we have introduced a new off-policy LSTD algorithm WIS-LSTD( ) that extends the benefits of WIS to reinforcement learning. Our empirical results show that the new WIS-LSTD( ) can outperform Yu?s off-policy LSTD( ) in both tabular and function approximation tasks and shows robustness in terms of its parameters. An interesting direction for future work is to extend these ideas to off-policy linear-complexity methods.

Acknowledgement This work was supported by grants from Alberta Innovates Technology Futures, National Science and Engineering Research Council, and Alberta Innovates Centre for Machine Learning.

8

2 References

Andrad?ottir, S., Heyman, D. P., Ott, T. J. (1995). On the choice of alternative measures in importance sampling with markov chains. Operations Research, 43(3):509?519. Bertsekas, D. P., Yu, H. (2009). Projected equation methods for approximate solution of large linear systems. Journal of Computational and Applied Mathematics, 227(1):27?50. Boyan, J. A. (1999). Least-squares temporal difference learning. In Proceedings of the 17th International Conference, pp. 49?56. Casella, G., Robert, C. P. (1998). Post-processing accept-reject samples: recycling and rescaling. Journal of Computational and Graphical Statistics, 7(2):139?157. Dann, C., Neumann, G., Peters, J. (2014). Policy evaluation with temporal differences: a survey and comparison. Journal of Machine Learning Research, 15:809?883. Geist, M., Scherrer, B. (2014). Off-policy learning with eligibility traces: A survey. Journal of Machine Learning Research, 15:289?333. Hachiya, H., Akiyama, T., Sugiayma, M., Peters, J. (2009). Adaptive importance sampling for value function approximation in off-

policy reinforcement learning. Neural Networks, 22(10):1399?1410. Hachiya, H., Sugiyama, M., Ueda, N. (2012). Importance-weighted least-squares probabilistic classifier for covariate shift adaptation with application to human activity recognition. Neurocomputing, 80:93?101. Hesterberg, T. C. (1988), Advances in importance sampling, Ph.D. Dissertation, Statistics Department, Stanford University. Kahn, H., Marshall, A. W. (1953). Methods of reducing sample size in Monte Carlo computations. In Journal of the Operations Research Society of America, 1(5):263?278. Koller, D., Friedman, N. (2009). Probabilistic Graphical Models: Principles and Techniques. MIT Press, 2009. Liu, J. S. (2001). Monte Carlo strategies in scientific computing. Berlin, Springer-Verlag. Maei, H. R., Sutton, R. S. (2010). GQ( ): A general gradient algorithm for temporal-difference prediction learning with eligibility traces. In Proceedings of the Third Conference on Artificial General Intelligence, pp. 91?96. Atlantis Press. Maei, H. R. (2011). Gradient temporal-difference learning algorithms. PhD thesis, University of Alberta. Precup, D., Sutton, R. S., Singh, S. (2000). Eligibility traces for off-policy policy evaluation. In Proceedings of the 17th International Conference on Machine Learning, pp. 759?766. Morgan Kaufmann. Precup, D., Sutton, R. S., Dasgupta, S. (2001). Off-policy temporal-difference learning with function approximation. In Proceedings of the 18th International Conference on Machine Learning. Robert, C. P., and Casella, G., (2004). Monte Carlo Statistical Methods, New York, Springer-Verlag. Rubinstein, R. Y. (1981). Simulation and the Monte Carlo Method, New York, Wiley. Shelton, C. R. (2001). Importance Sampling for Reinforcement Learning with Multiple Objectives. PhD thesis, Massachusetts Institute of Technology. Shimodaira, H. (2000). Improving predictive inference under covariate shift by weighting the log-likelihood function. Journal of Statistical Planning and Inference, 90(2):227?244. Sutton, R. S., Barto, A. G. (1998). Reinforcement Learning: An Introduction. MIT Press. Sutton, R. S., Mahmood, A. R., Precup, D., van Hasselt, H. (2014). A new Q( ) with interim forward view and Monte Carlo equivalence. In Proceedings of the 31st International Conference on Machine Learning, Beijing, China. Yu, H. (2010). Convergence of least squares temporal difference methods under general conditions. In Proceedings of the 27th International Conference on Machine Learning, pp. 1207?1214.

9