TREE-BACKUP BIAS-VARIANCE TRADE-OFFS

David Krueger

ABSTRACT

Building on Kearns & Singh (2000), we perform a bias-variance analysis of (on-policy) Tree Backup (TB) policy evaluation. While the same argument of Kearns & Singh (2000) applies to SARSA, we show that (for soft-policies) TB has higher bias (and lower variance), since it gives more weight to value-function estimates and less weight to observed rewards. This observation can help explain the advantage of SARSA over TB in early stages of learning, as demonstrated in a previous assignment.

1 INTRODUCTION

First, since the update targets for (Expected) SARSA(n) are the same as for TD(n) [1], the argument of Kearns & Singh (2000) can be straightforwardly generalized; all that is needed is to have l trajectories starting in every state-action (as opposed to just every state in Kearns & Singh (2000)). Similarly, Δ_t would be used to represent the maximum error in the estimate of the Q-function.

2 BIAS AND VARIANCE OF TREE BACKUPS

We use the same basic technique of Kearns & Singh (2000) to analyze the TB algorithm:

1. Show that the TB update rule can be derived as a sample based estimation of q^π expanded using the Bellman equation
2. Examine the bias/variance of the bootstrapping/sampling estimates of the terms in the expansion of q^π.

2.1 STEP 1: TREE BACKUPS VIA BELLMAN EXPANSIONS

The Bellman equation for q^π can be used to recursively expand $q^\pi(s)$, and we can view algorithms such as TB as approximating the elements of the expansion. Defining $v_\pi(s) \doteq \mathbb{E}_{\pi(a|s)} q_\pi(s,a)$, the equation is:

$$q^\pi(s_1, a_1) = \mathbb{E}_{P(r,s_2|s,a)} r + \gamma v_\pi(s_2) \tag{1}$$

We show the first three steps of the expansion which corresponds to the TB algorithm. For TB(n), we would continue expanding the trailing q_k term a total of n times. We now subsume the expectations into an expectation over trajectories (ζ) beginning with s_1, a_1. For simplicity, we drop the π subscripts of q and v, and use subscripts of r, q, and v to indicate which state(-action) they are being evaluated at, e.g. $v_2 \doteq v_\pi(s_2)$. We also introduce c_k as arbitrary constants, following the notation of Munos et al. (2016).

$$\begin{aligned}
q_1 &= (q_1 - q_1) + \mathbb{E}_{\zeta \sim P(\zeta|\pi,P,s_1,a_1)}[r_2 + \gamma v_2] && (2)\\
&= (q_1 - q_1) + \mathbb{E}_{\zeta \sim P(\zeta|\pi,P,s_1,a_1)}[r_2 + \gamma v_2 - c_2(\gamma q_2 + \gamma q_2)] && (3)\\
&= (q_1 - q_1) + \mathbb{E}_{\zeta \sim P(\zeta|\pi,P,s_1,a_1)}[r_2 + \gamma v_2 - \gamma c_2 q_2 + \gamma c_2(r_3 + \gamma v_3 - c_3(\gamma q_3 + \gamma q_3))] && (4)
\end{aligned}$$

[1] Kearns & Singh (2000) calls the algorithm TD(k), but we use TD(n), following Sutton & Barto (2017); we further use l where they use n.

Here we see that, unlike in SARSA(n), which only bootstraps at the last time-step, we maintain components of the value function at every time-step in our expression for q. This is significant because bootstrapping causes bias (Kearns & Singh, 2000).

We can move q_1 inside the expectation, since it doesn't depend on any of $r_1, s_2, a_2, ...$, and regroup terms:

$$q_1 = q_1 + \mathbb{E}_{\zeta \sim P(\zeta|\pi,P,s_1,a_1)}[(-q_1 + r_2 + \gamma v_2) + \gamma c_2(-q_2 + r_3 + \gamma v_3) + \gamma^2 c_2 c_3(-q_3 + q_3)] \tag{5}$$

Now we recognize Equation 7.12 of Sutton & Barto (2017) as estimating the terms which make up this expansion of the Q-function when we set $c_1 = 1$ and $c_k = \pi(a_k|s_k)$ for $k > 1$ (although we emphasize that the above derivation does not depend on this in any way).

2.2 Step 2: Bias-Variance Analysis

We now rewrite Equation 7.12 of Sutton & Barto (2017) using the notation $d_k \doteq \prod_{i=1}^{k} \pi(A_i|S_i)$, and replacing $\min(n, T-1)$ with n (by treating termination as inhabiting the terminal state). We leave out the superscipt, but note that V and Q are estimates of the value function of policy π. We then expand the δ_k term as $\delta_k = R_{k+1} + \gamma V_{k+1} - Q_{k-1}(S_k, A_k)$, in order to separate the components corresponding to value functions vs. rewards.

$$G_1^{(n)} = Q_0(S_1, A_1) + \sum_{k=1}^{n} \delta_k \prod_{i=2}^{k} \gamma \pi(A_i|S_i) \tag{6}$$

$$= Q_0(S_1, A_1) + \sum_{k=1}^{n} \delta_k \gamma^{k-1} d_k \tag{7}$$

$$= Q_0(S_1, A_1) + \sum_{k=1}^{n} \gamma^{k-1} d_k R_{k+1} + \sum_{k=1}^{n} \gamma^{k-1} d_k (\gamma V_{k+1} - Q_{k-1}(S_k, A_k)) \tag{8}$$

$$= Q_0(S_1, A_1) + \sum_{k=1}^{n} \gamma^{k-1} d_k R_{k+1} + \sum_{k=1}^{n} \gamma^{k-1} d_{k-1} (V_k - c_k Q_{k-1}(S_k, A_k)) + \gamma^n d_n V_{n+1} \tag{9}$$

When the policy is deterministic, $c_k = d_k = 1$ always, and $V_k = c_k Q_{k-1}(S_k, A_k)$, so all the value-function terms cancel except the final $\gamma^n V_{n+1}$, and TB and SARSA become equivalent. On the other hand, if we assume that π is an ϵ-soft policy, so $c_k \leq 1 - (|\mathcal{A}| - 1)\epsilon$, then the TB targets give less weight to the rewards terms and include more value terms.

2.2.1 Variance Term

So now we use this bound on c_k to get a tighter bound on the variance of the reward terms as estimates of the true expected reward (with probability δ).

Defining $\beta \doteq (1 - (|\mathcal{A}| - 1)\epsilon)$ and $\tilde{\gamma} \doteq \beta\gamma$, and using the same large deviation analysis as in Kearns & Singh (2000), we get the variance term:

$$\frac{1 - \tilde{\gamma}^n}{1 - \tilde{\gamma}} \sqrt{\frac{3 \log(n/\delta)}{l}} \tag{10}$$

Which is (as expected) less than the variance of SARSA, since $\tilde{\gamma} < \gamma$.

2.2.2 Bias Term

Turning to the value function terms, we define $\Delta_t \doteq \max_{s,a} |Q_t(s,a) - q^\pi(s,a)|$ every Q_t term contributes Δ_t to the bias component of the probabilistic bound on Δ_{t+1}. Considering the coefficients

of these terms, we note that d_{k-1} will be *larger* and $(V_k - c_k Q_{k-1}(S_k, A_k))$ will be *smaller* when more likely actions were chosen up to time-step k.

The sum of these coefficients is:

$$\sum_{k=1}^{n} \gamma^{k-1} d_{k-1}(1 - c_k) = (1 - c_1) + \gamma c_1(1 - c2) + \gamma^2 c_1 c_2(1 - c_3) + ... \tag{11}$$

$$= 1 - c_1 + \gamma c_1 - \gamma c_1 c2 + \gamma^2 c_1 c_2 - \gamma^2 c_1 c_2 c_3 + ... \tag{12}$$

$$= 1 + (\gamma - 1)c_1 + \gamma(\gamma - 1)c_1 c2 - \gamma^2 c_1 c_2 c_3 + ... \tag{13}$$

$$= 1 - (1 - \gamma)\left(\sum_{k=1}^{n-1} \gamma^{k-1} \prod_{i=1}^{k} c_i\right) - \gamma^{n-1} \prod_{i=1}^{n} c_i \tag{14}$$

Since all of the terms involving c_i are negative and linear in c_i, maximizing this expression just amounts to minimizing all of the c_i. By hypothesis, π is ϵ-soft, and so the minimum value of c_i is just ϵ. Plugging in $c_k = \epsilon$ Into the original expression (left side of Equation 11) yields:

$$\sum_{k=1}^{n} \gamma^{k-1} d_{k-1}(1 - c_k) = (1 - \epsilon) + \sum_{k=2}^{n} \gamma^{k-1} \epsilon^{k-2}(1 - \epsilon) \tag{15}$$

$$= (1 - \epsilon) + \gamma(1 - \epsilon) \sum_{k=0}^{n-2} (\gamma\epsilon)^k \tag{16}$$

$$= (1 + \gamma)(1 - \epsilon)\frac{1 - (\gamma\epsilon)^{n-1}}{1 - \gamma\epsilon} \tag{17}$$

2.2.3 Combining Terms and Solving the Recurrence

Now the total bound on Δ_{t+1} is:

$$\Delta_{t+1} \leq \frac{1 - \tilde{\gamma}^n}{1 - \tilde{\gamma}} \sqrt{\frac{3 \log(n/\delta)}{l}} + \left((1 + \gamma)(1 - \epsilon)\frac{1 - (\gamma\epsilon)^{n-1}}{1 - \gamma\epsilon} + \gamma\tilde{\gamma}^{n-1}\right) \Delta_t. \tag{18}$$

Solving this recurrence (under the assumption that $\Delta_0 = 1$) gives:

$$\Delta_t \leq \frac{1 - \xi^t}{1 - \xi} \left(\frac{1 - \tilde{\gamma}^n}{1 - \tilde{\gamma}} \sqrt{\frac{3 \log(n/\delta)}{l}}\right) \tag{19}$$

with

$$\xi \doteq (1 + \gamma)(1 - \epsilon)\frac{1 - (\gamma\epsilon)^{n-1}}{1 - \gamma\epsilon} + \gamma\tilde{\gamma}^{n-1} \tag{20}$$

Both terms of the expression are positive, and the first term is strictly *increasing* with n, while the seconds term is similar to the term in SARSA, and decreases as a function of n. This means that (unlike with SARSA(n)), there is a positive lower bound to the bias term of the bound for the Tree Backup algorithm, suggesting that the extra bootstrapping of TB leads *unavoidably* to some minimum amount of bias in the updates.

Note that the bound we've provided may or may not be strong enough to guarantee convergence of the Δ_t. For instance, if $n = 2, \gamma = 1/2, \epsilon < 1/3$, then the first term alone is greater than 1, and so the bound will diverge. But for $n = 2, \gamma = 1/2, \epsilon = 1/2$, we have $\xi = \frac{3}{2}\frac{1}{2} + \frac{1}{2}\frac{1}{2}\frac{1}{2} = \frac{7}{8} < 1$, and so it does converge.

3 Future Work and Conclusions

We've derived a bound for the error in the Q-function after t phased updates using (on-policy) TB(n). Unfortunately, the bound does not always guarantee convergence.

The variance component of this bound is less than for SARSA(n), and we suspect but do not show that the reduction in variance of TB(n) must be compensated for by increased bias. We do find the bias term is bounded below by a positive constant.

There are several ways in which the analysis here could be improved. Firstly, we considered the worse case values of c_i independently for each term of bound (Equation 18), which is suboptimal. In fact, increasing c_i increases the first and third terms of the bound, but decreases the second term.

A next step could be considering probabilistic bounds on Equation 11 (instead of the worst-case analysis we performed). The bound on Δ_t is already probabilistic, so it seems unlikely that using a worst-case analysis here always yields the optimal bound. Also note that the worst case (for the 2nd term of Equation 18) corresponds to (one of) the least probable action(s) being chosen at every time-step, so in practice we'd expect this part of the bound to usually be quite loose.

It would also be interesting to use our bound (or similar) to derive a theoretical scheduling of the σ parameter in $Q(\sigma)$ following Kearns & Singh (2000) and compare this with the schedules used by De Asis* et al. (2017).

REFERENCES

K. De Asis*, J. F. Hernandez-Garcia*, G. Zacharias Holland*, and R. S. Sutton. Multi-step Reinforcement Learning: A Unifying Algorithm. March 2017.

Michael J. Kearns and Satinder P. Singh. Bias-variance error bounds for temporal difference updates. In *Proceedings of the Thirteenth Annual Conference on Computational Learning Theory*, COLT '00, pp. 142–147, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc. ISBN 1-55860-703-X. URL http://dl.acm.org/citation.cfm?id=648299.755183.

Rémi Munos, Tom Stepleton, Anna Harutyunyan, and Marc G. Bellemare. Safe and efficient off-policy reinforcement learning. *CoRR*, abs/1606.02647, 2016. URL http://arxiv.org/abs/1606.02647.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning : An Introduction, 2nd edition*. MIT Press, DRAFT, 2017. URL http://incompleteideas.net/sutton/book/bookdraft2016sep.pdf.