Manipulating the action formulation of an RL agent is certainly not the only way to use Lyapunov domain knowledge. Ng et al. (1999) proposed a general method for changing the reward function of an MDP to encourage movement toward a goal as measured by some "potential" function. Lyapunov functions seem excellent candidates for the potential functions in such a scheme. Combining our approach with their might yield even better results. Alternatively, in a Bayesian framework such as the one proposed by Dearden et al. (1998), Lyapunov domain knowledge might be used to set priors about the values of actions or about which actions are more likely to be optimal ones.

Acknowledgments

This work was funded by the National Science Foundation under Grant No. ECS-9808002 and ECS-0070102. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. Theodore Perkins is also supported by a graduate fellowship from the University of Massachusetts Amherst. We thank Doina Precup and Michael Rosenblum for comments on earlier drafts of this paper.

References


Dietterich, T. G. Hierarchical reinforcement learning with the max-value function decomposition. Journal of Artificial Intelligence Research, 13, 227-303.


Abstract

We introduce the first algorithm for off-policy temporal-difference learning that is stable with linear function approximation. Off-policy learning is of interest because it forms the basis for popular reinforcement learning methods such as Q-learning, which has been known to diverge with linear function approximation, and because it is critical to the practical utility of multi-scale, multi-goal, learning frameworks such as options, HAMs, and MAXQ. Our new algorithm assigns TD(λ) over state-action pairs with importance sampling idea from our previous work. We prove that, given training under any ε-soft policy, the algorithm converges w.p.1 to a close approximation (as in Tsitsiklis and Van Roy, 1997; Tadic, 2001) to the action-value function for the target policy. Variations of the algorithm designed to reduce variance introduce additional bias but are also guaranteed convergent. We also illustrate our method empirically on a small policy-evaluation problem, showing reduced variance compared to the most obvious importance sampling algorithm for this problem. Our current results are limited to episodic tasks with episodes of bounded length.

Although Q-learning remains the most popular of all reinforcement learning algorithms, it has been known since about 1999 that it is unsound with linear function approximation (see Gordon, 1995; Bertsekas and Tsitsiklis, 1996). The most telling counterexample, due to Baird (1995), is a seven-state Markov decision process with linearly independent feature vectors for which an exact solution exists, yet for which the approximate values found by Q-learning diverge to infinity. This problem prompted the development of residual gradient methods (Baird, 1995), which are stable but much slower than Q-learning, and fitted value iteration (Gordon, 1995, 1999), which is also stable but limited to restricted, weaker-than-linear function approximators. Of course, Q-learning has been used with linear function approximation since its invention (Watkins, 1989), often with good results, but the soundness of this approach is no longer an open question. There exist non-pathological Markov decision processes for which it diverges; it is absolutely unsound in this sense. A sensible response is to turn to some of the other reinforcement learning methods, such as SARSA, that are also efficient and for which soundness remains a possibility. An important distinction here is between methods that must follow the policy they are learning about, called on-policy methods, and those that can learn from behavior generated by a different policy, called off-policy methods. Q-learning is an off-policy method in that it learns the optimal policy even when actions are selected according to a more exploratory or even random policy. Q-learning requires only that all actions be tried in all states, whereas on-policy methods like SARSA require that they be selected with specific probabilities.

Although the off-policy capability of Q-learning is appealing, it is also the source of at least part of its instability problems. For example, in one version of Baird's counterexample, the TD(λ) algorithm, which underlies both Q-learning and SARSA, is applied with linear function approximation to learn the action-value function $Q^*$ for a given policy $\pi$. Operating in an on-policy mode, updating state-action pairs according to the same distribution with which they would be experienced under $\pi$, this method is stable and convergent near the best possible solution (Tsitsiklis and Van
Given a state $s$, $0 < t < T$, the action $a_{t}$ is selected according to probability $\pi(s_{t}, a_{t})$ or $\pi(s_{t}, o_{t})$ depending on whether policy $\pi$ or policy $b$ is in force. We always use $\pi$ to denote the target policy, the policy that we are learning about. In the on-policy case, $\pi$ is also used to generate the actions of the episode. In the off-policy case, the actions are instead generated by $b$, which we call the behavior policy.

In either case, we seek an action that maximizes the action-value function $Q^{*} : S \times A \rightarrow \mathbb{R}$ for the target policy $\pi$:

$$Q^{*}(s, a) = E_{\tau \sim \tau^{(T)}} \left[ \sum_{t=0}^{T-1} \gamma^{t} r_{t+1} \mid s_{0} = s, a_{0} = a \right],$$

where $0 \leq \gamma \leq 1$ is a discount-rate parameter.

We consider approximations that are linear in a set of feature vectors $\phi_{a}(s, a) \in \mathbb{R}^{d}$, where $\theta \in \mathbb{R}^{d}$ is the learned parameter vector. The feature vector for the special terminal state is assumed to be the zero vector so that the estimated value for this state is (correctly) zero.

In this paper we restrict our attention to per-episode updating, meaning that although an increment to $\theta$ is computed on each episode, $\theta$ is not actually updated until the end of the episode (by a total increment, $\Delta \theta$, equal to the sum of the increments on each episode). The increments for conventional TD(1) under per-episode updating are given by the forward-view equations:

$$\Delta \theta = \alpha \left( R_{T} - \theta^{\top} \phi_{a_{T}} \right) \phi_{a_{T}}$$

where $R_{T} = r_{T+1} + \gamma r_{T+2} + \cdots + \gamma^{T-1} r_{T+T}$.

We take the initial state and action, $s_{0}$ and $a_{0}$, to be given arbitrarily. Given a state and action, $s_{t}$ and $a_{t}$, the next reward, $r_{t+1}$, is a random variable with mean $\mathbb{E}[r_{t+1}]$. In the case of $a_{t}$, a trajectory of state-action pairs is initiated. The terminal state is a special terminal state that may not occur on any preceding time step.

1. Notation and Main Result

We consider the standard episodic reinforcement learning framework (see, e.g., Sutton & Barto, 1998) in which a learning agent interacts with a Markov decision process (MDP). Our notation focuses on a single episode of $T$ time steps, $s_{0}, a_{0}, r_{1}, s_{1}, a_{1}, r_{2}, \ldots, r_{T}, s_{T}$, with states $s \in S$, actions $a \in A$, and rewards $r \in \mathbb{R}$.

The most straightforward way to introduce importance sampling into linear TD(1) is to multiply the increments for each episode by the relative probability of that episode occurring under the target and behavior policies. If we define the importance sampling ratio as $r_{ts} = r_{ts}^{(b)} / r_{ts}^{(T)}$, then this relative probability is $p_{b} / p_{T}$. Let us write this as the noisy importance

$$\Delta \theta_{b} = \alpha \left( R_{T} - \theta^{\top} \phi_{a_{T}} \right) \phi_{a_{T}}$$

where $R_{T} = r_{T+1} + \gamma r_{T+2} + \cdots + \gamma^{T-1} r_{T+T} + \gamma^{T} \mathbb{E}[r_{T+T}]$.

2. Convergence and Error Bounds

Given Theorem 1, we can apply the analysis of Tsitsiklis and Van Roy to prove convergence and error bounds. Their paper (Tsitsiklis & Van Roy, 1997) treated the discounted continuing (ergodic) case, whereas here we consider the episodic case. Their results for this case were published in the textbook by Bertsekas and Tsitsiklis (1995). Gurvits also obtained similar results, and some of the ideas can be traced back to his work (Gurvits, Lin & Hansen, unpublished). Tadic (2001) proved a similar result using different mathematical techniques, and a less restrictive set of assumptions.

Let $\delta_{s} : S \times A \rightarrow [0, 1]$ where $\delta_{s}(s, a)$ is the (arbitrary) distribution of starting state-action pairs. Let $P_{b}$ be the state-action pair to state-action pair transition-probability matrix for policy $b$. Then $P_{b} = \sum_{s,a} \delta_{s}(s, a) b(s, a)$.

We require a number of natural assumptions: (1) the state and action sets are finite; (2) all state-action
Given a state, $s_t$, $0 \leq t < T$, the action $a_t$ is selected according to probability $\pi(s_t, a_t) = \theta(s_t, a_t)$ depending on whether policy $\pi$ or policy $b$ is in force. We always use $\pi$ to denote the target policy, the policy that we are learning about. In the on-policy case, $\pi$ is also used to generate the actions of the episodes. In the off-policy case, the actions are instead generated by $b$, which we call the behavior policy.

In either case, we seek an action-value function $Q^\pi: S \times A \to \mathbb{R}$ for the target policy $\pi$:

$$Q^\pi(s, a) = E_{\tau_{s,a}} \left[ \sum_{t=0}^{\infty} \gamma^t R_t | s_0 = s, a_0 = a \right],$$

where $0 \leq \gamma \leq 1$ is a discount rate parameter. We consider approximations that are linear in a set of feature vectors $\phi_1, \ldots, \phi_M \in \mathbb{R}^d$, $d > 0$,

$$Q^\pi(s, a) = \theta^T \phi(s, a),$$

where $\theta \in \mathbb{R}^M$ is the learned parameter vector. The feature vector for the special terminal state is assumed to be the zero vector so that the estimated value for this state is (correctly) zero.

In this paper we restrict our attention to per-episode updating, meaning that although an increment to $\theta$ is computed on each episode, $\theta$ is not actually updated until the end of the episode (by a total increment, $\Delta \theta$, equal to the sum of the increments on each episode). The increments for conventional TD($\lambda$) under per-episode updating are given by the forward-view equations:

$$\Delta \theta_t = \alpha \left( R_{t+1} - \theta^T \phi_t \right) \phi_t,$$

where $R_{t+1}$ is a shorthand for $R_{t+1}^\pi$, with $\phi_t = \phi(s_t, a_t)$, $t = 0, \ldots, T - 1$.

The off-policy n-step return was introduced by Precup, Sutton and Singh (2000) as part of their per-episode importance sampling algorithm. They showed that the importance sampling ratios correct for off-policy training such that

$$E_\tau \left[ R^\pi_t | s_t = a_t, \alpha_t \right] = E_\tau \left[ R^\pi_t | s_t = a_t \right],$$

where the subscript on the expectations indicates the policy in force (i.e., they indicate either off-policy training, $b$, or on-policy training, $\pi$,). Here we extend this idea to the case of linear function approximation by including the correction ratios in (1). We are now ready to state our main result:

**Theorem 1** Let $\Delta \theta$ and $\Delta \theta^\pi$ be the sum of the parameter increments over an episode under on-policy TD($\lambda$) and importance sampled TD($\lambda$), respectively, assuming that the starting weight vector is $\theta$ in both cases. Then

$$\theta_t \in S, a_t \in A, \forall t \in \mathbb{N}, \alpha_t \in \mathcal{A}.$$
pairs are visited under the behavior policy from $d_t$, both behavior and target policies, $\pi$ and $\tilde{\pi}$, are proper, meaning that $P_{\pi} = P_{\tilde{\pi}} = 0$; (4) the rewards are bounded; and (5) the step-size sequence $(\alpha_k)$ satisfies the usual stochastic approximation conditions:

$$\alpha_k \geq 0, \sum_{k=0}^{\infty} \alpha_k = \infty, \text{ and } \sum_{k=0}^{\infty} \alpha_k^2 < \infty. \quad (2)$$

In addition, we require: (6) the variance of the product of correction factors be bounded for any initial state:

$$E_\rho(\tilde{p}_1 \tilde{p}_2 \ldots \tilde{p}_k) < B \quad \forall \rho \in \mathcal{S},$$

which can be assured, for example, by simply bounding the possible episode lengths. Nevertheless, this remains a limitation of our result, as discussed further below. Lastly, let $Q_\rho$ denote the approximate action–value function (vector) for any parameter value $\rho$: $Q_\rho(s, a) = \tilde{\pi}_\rho(s, a)$.

Theorem 3 Under the assumptions 1–6 above, the importance-sampled TD(\lambda) converges with probability one to some $w_\lambda$ such that

$$||Q_{w_\lambda} - Q^*||_1 \leq \min_{\rho \in \mathcal{S}} ||Q_\rho - Q^*||_1 \left( 1 - \alpha \right),$$

where $\beta$ is the contraction factor of the matrix

$$M = (1 - \lambda) \sum_{k=0}^{\infty} \lambda^k (\gamma P_{\tilde{\pi}})^k.$$

Proof: This result is a restatement of Taittsiklis and Van Roy’s result on pages 312 of Bertsekas and Tsitsiklis (1996). The assumptions together with our main result immediately satisfy the conditions and their proof. In particular, assumption 6 implies that our importance sampling corrections do not convert the usual estimator to one of bounded variance.

The assumption of bounded variance of the correction-factor product (6) is restrictive, but not as restrictive as it might seem at first. In many cases we can assure its satisfaction by considering only “artificial” episode terminations superimposed on an original process. For example, assume assumption 6 is trivially met if the trial length is bounded. Even if our original MDP does not produce bounded length trials, we can consider a modified MDP that is just like the original except that all trials terminate after $T_{\text{MAX}}$ steps. Sample trajectories from the original process can be used as trajectories for the modified process by truncating them after $T_{\text{MAX}}$ steps. Our results assure stable convergence to a close approximation to the true evaluation function for the modified MDP and, if $T_{\text{MAX}}$ is chosen large enough compared to $\gamma$ or the mixing time of the original MDP, then the solutions to the original and modified MDPs will be very similar.

In our primary expected application area—learning about temporally abstract macro-actions—a kind of artificial termination is the normal way of proceeding. A macro-action consists of a target policy and a specified condition for terminating the macro-action. In this application it is not the actual process that terminates, only the macro-action. Nevertheless, the problem is formally identical to the one presented in this paper; our methods and results apply directly to learning about macro-actions. And in fact, choosing the termination process is part of designing the macro-action. Thus we can design the macro-actions to have bounded variance of the correction term by terminating after $T_{\text{MAX}}$ steps, for example, or whenever the correction factor becomes very large.

Thus, in many applications, the rates of divergence due to unbounded variance can be eliminated. Nevertheless, even when bounded, high variance (and thus slow convergence) can be a major problem. In Section 6 we consider how weighted importance sampling methods might be adapted to reduce variance, or even remove the need for assumption 6.

3. Restarting within an Episode

The importance-sampling correction product is (1) will often decay very rapidly over time, especially if the episodes are long or if the behavior and target policy are very different. Although the episode may be continuing, little more is learned once the correction factor becomes very small. In such cases one might like to pretend a new episode has started from an initial state of the episode. Of course, the effective starting distribution will then be different from $\tilde{\pi}$, which might be considered to introduce additional bias. Nevertheless, this may be desirable because of reduced variance. In this section we prove convergence of this generalised algorithm.

To formalise the idea of starting anywhere within an episode, we introduce a non-negative random variable $g_t$, which is allowed to depend only on events up to (and including) time $t$. The value $g_t$ represents the extent to which an episode is considered to start at time $t$. The function $g: \{0, 1\} \rightarrow \mathbb{R}^+$ gives the expected value of $g_t$ for any trajectory up through $t$. The forward view of the generalised algorithm is

$$\Delta \theta_t = \alpha \left( R_t - \bar{\theta}_t \right) + \lambda \sum_{k=0}^{\infty} \gamma^k \tilde{p}_t \tilde{p}_{t+1} \cdots \tilde{p}_k g_{t+k} \quad (3)$$

Note that this algorithm is identical to the original importance-sampling TD(\lambda) if $g_t = 1$ and $g_t = 0, \forall t \neq 1$.

Theorem 3 Let $\Delta \theta$ and $\tilde{\Delta} \theta$ denote the sum of the parameter increments of the original importance-sampled TD(\lambda) (1) and the generalised version (3) respectively. Then, for any function $g$, there exists an alternate starting distribution $d'_{\theta}$ such that

$$E_{\rho}(\Delta \theta | s, a, s') = E_{\rho}(\tilde{\Delta} \theta | s, a, s') + d'_{\theta}.$$

Proof: To simplify notation, we allow additional subscripts on the expectations to indicate the distribution from which the initial $s_0, a_0$ are selected. Then

$$E_{\rho}(\tilde{\Delta} \theta) = E_{\rho}(\tilde{\Delta} \theta_{s_0, a_0}),$$

where $\rho_{s_0, a_0}$ denotes the set of all trajectories $\omega$ of length $k$ and $\tilde{\rho}(\omega)$ denotes the probability of each such trajectory occurring under $\tilde{\pi}$ when starting from $d$. The final expectation above is conditional on starting in the indicated last state and action $s_0, a_0$ of $\omega$. It is convenient now to define $\Omega_{s_0, a_0}$ as the set of all trajectories of length $k$ ending is $s_0, a_0$. Then we can rewrite the above as

$$\sum_{s_0, a_0} \sum_{\omega \in \Omega_{s_0, a_0}} \tilde{\rho}(\omega) E_{\rho}(\tilde{\Delta} \theta_{s_0, a_0, \omega}) = \Delta \theta_{s_0, a_0},$$

where $\Delta \theta_{s_0, a_0}$ is given by (2).

3. At the end of the episode

$$\theta_t = - \sum_{s} \Delta \theta_s t$$

Figure 1. Incremental implementation of importance-sampled TD(\lambda)

Theorem 4 The backward-view description given in Figure 1 is equivalent to the forward-view definition (3).

Proof: From the algorithm definition,

$$c_t = \sum_{a_0} \tilde{p}_t g_{t+k} \quad (4)$$

Therefore, in the forward view, we can re-write the sum of the updates that occur during an episode as

$$\sum_{s_0, a_0} \alpha \left( R_{s_0, a_0} \right) \sum_{t=1}^{T} \tilde{p}_{s_0, a_0+t} \cdots \tilde{p}_t g_{s_0, a_0+t} \cdots g_t \quad (5)$$

is clearly a valid alternative starting distribution.
Theorem 3 Under the assumptions 1-6 above, episodic importance sampled TD(λ) converges with probability one to some \( f_w \) such that

\[
||Q_{\text{est}} - Q||^2 \leq \min_{\beta} \left( ||Q_{\text{est}} - Q|| \right) \frac{1}{1 - \beta^2}
\]

where \( \beta \) is the contraction factor of the matrix

\[
M = (1 - \lambda) \sum_{k=0}^{\infty} \lambda^k (\gamma P_k)^{k+1}
\]

Proof: This result is a restatement of Tatsitsikis and Van Roy’s result on pages 312 of Bertsekas and Tsitsiklis (1996). The assumptions together with our main result immediately satisfy the conditions of their proof. In particular, assumption 6 implies that our importance sampling corrections do not convert the usual estimator to one of unbounded variance.

The assumption of bounded variance of the correction-factor product \( \rho \) is restrictive, but not as restrictive as it might seem at first. In many cases we can assure its satisfaction by considering only “artificial” episodic terminations superimposed on an original process. For example, an assumption 6 is trivially met if the trial length is bounded. Even if our original MDP does not produce bounded length trials, we can consider a modified MDP that is just like the original except that all trials terminate after \( T_{\text{max}} \) steps. Sample trajectories from the original process can be used as trajectories for the modified process by truncating them after \( T_{\text{max}} \) steps. Our results assure stable convergence to a close approximation to the true evaluation function for the modified MDP and, if \( T_{\text{max}} \) is chosen large enough compared to \( T \) or the mixing time of the original MDP, then the solutions to the original and modified MDPs will be very similar.

In our primary expected application area—learning about temporally abstract macro-actions—this kind of artificial termination is the normal way of proceeding. A macro-action consists of a target policy and a specified condition for terminating the macro-action. In this application it is not the actual process that terminates, only the macro-action. Nevertheless, the problem is formally identical to the one presented in this paper; our methods and results apply directly to learning about macro-actions. And in fact, choosing the termination process is part of designing the macro-action. Thus we can design the macro-actions to have bounded variance of the correction term by terminating after \( T_{\text{max}} \) steps, for example, or whenever the correction factor becomes very large.

Thus, in many applications, the sparsity of divergence due to unbounded variance can be eliminated. Nevertheless, even when bounded, high variance (and thus slow convergence) can be a major problem. In Section 6 we consider how weighted importance sampling methods might be adapted to reduce variance, or even remove the need for assumption 6.

3. Restating within an Episode

The importance sampling correction product in (1) will often decay very rapidly over time, especially if the episodes are long or if the behavior and target policy are very different. Although the episode may be continuing, little more is learned once the correction factor becomes very small. In such cases one might like to pretend a new episode has started from an intermediate state of the episode. Of course, the effective starting distribution will then be different from \( \pi \), which might be considered to introduce additional bias. Nevertheless, this may be desirable because of reduced variance. In this section we prove convergence of this generalised algorithm.

To formalise the idea of starting anywhere within an episode, we introduce a non-negative random variable \( g_t \) which is allowed to depend only on events up to (including) time \( t \). The value of \( g_t \) represents the extent to which an episode is considered to start at time \( t \). The function \( g : \mathbb{N} \rightarrow \mathbb{R}^+ \) gives the expected value of \( g_t \) for any trajectory up through \( t \). The forward view of the generalised algorithm is

\[
\Delta \theta_t = \alpha (R_t - \theta_t \phi_t) \phi_{t+1} + \sum_{k=0}^{\infty} g_{t+k} (\lambda P_k \theta_{t+k+1} + \cdot \cdot \cdot)
\]

Note that this algorithm is identical to the original importance sampled TD(\( \lambda \)) if \( g = 1 \) and \( g = 0, \forall \lambda \geq 1 \).

Theorem 4 Let \( \bar{D} \) and \( \bar{D}^\delta \) denote the sum of the parameter increments of the original importance-sampled TD(\( \lambda \)) (1) and the generalised version (3) respectively. Then, for any function \( g \), there exists an alternate starting distribution \( d_0 \) such that

\[
E_g[Q(\bar{D})] = E_g[Q(\bar{D}^\delta)]
\]

Proof: To simplify notation, we allow additional subscripts on the expectations to indicate the distribution from which the initial \( d_0 \) is selected. Then

\[
E_g[Q(\bar{D})] = \sum_{s} \sum_{a} \sum_{o} \sum_{r} \sum_{s'} \rho(s,a,o,r,s') E_g[\phi(s)]
\]

3. At the end of the episode, \( \theta \rightarrow \theta + \sum \Delta \theta_t \)

We have just proved that restarting in a general way, at any point during an episode, is equivalent to a conventional at-the-beginning starting distribution. The latter case we have already proved to converge; thus so must the generalised algorithm. The only difference is that the value converged to now depend on \( g \), and thus on \( \lambda \), rather than on \( \lambda \) and \( \pi \) alone.

4. Incremental implementation

The algorithm presented in the previous section can easily be implemented in an incremental, backward-view fashion, using an eligibility trace vector \( E_t \) of the same dimension as \( \theta \). This implementation, which we used in the experiments that follow, is given in Figure 1.

On every episode:

1. Initialize \( d_0 = \pi_0, E_0 = 0_0 \).

2. On every transition \((s,a) \rightarrow (s',a',r,q_{a'|s'})\), for \( 0 \leq t < T \):

\[
\rho_{s,a} = \pi(s,a)/\pi(s,a'), \quad \delta_t = r_{t+1} + \gamma \pi_{s',a'} - r_t - \gamma \theta_t
\]

\[
\Delta \theta_t = \alpha \delta_t \theta_t
\]

\[
E_{t+1} = E_t + \rho_{s,a} \delta_t \theta_t
\]

3. At the end of the episode, \( \theta \leftarrow \theta + \sum \Delta \theta_t \)

\[
\sum_{t=0}^{T-1} \alpha (R_t - \theta_t \phi_t) \phi_{t+1}
\]

Therefore, in the forward view, we can rewrite the sum of the updates that occur during an episode as:

\[
\sum_{t=0}^{T-1} \alpha (R_t - \theta_t \phi_t) \phi_{t+1} = \sum_{t=0}^{T-1} (R_t - \theta_t \phi_t) \phi_{t+1}
\]
5. An Empirical Illustration

To illustrate our algorithm we use the 11 x 11 grid-world environment depicted in Figure 2. The MDP is deterministic and has 4 actions, moving up, down, left or right. If the agent bumps into a wall, it remains in the same state. The four corner states are terminal. The agent receives a reward of R = 1 for the actions entering the bottom-right and upper-left corners, and R = -1 for entering the other two corners. All other rewards are 0. The initial state is the center, and the initial action is chosen randomly to be right or left. The target policy chooses down 40% of the time and up 10% of the time, with right and left chosen 25% of the time. The behavior policy is similar except with reversed up/down probabilities; it chooses down 10% of the time and up 40% of the time. In order to ensure that all the conditions of our convergence theory hold, trials are limited to 1000 time steps. However, this upper limit was never reached during our experiments.

The features used by the function approximator are overlapping stripes of width 3, parallel to the vertical axis. There are 13 such stripes. One consequence is that under the target policy, actions from the leftmost column have negative value, whereas all actions from the rightmost column have negative values. The situation is reversed under the behavior policy.

We implemented the incremental (backward view) version of importance sampled TD( ) with = 0 and = 0.9, and updates taking place only at the end of an episode. Because the results are very similar, we only present the data for = 0.9. The initial parameter of the function approximator was 0 = 0.

6. On the Possibility of Weighted Importance Sampling Methods

We have introduced a new, off-policy version of linear TD() and shown that it converges near the best solution consistent with its structure. However, excessive variance remains an issue, and there may be algorithms that reach the same asymptotic solution faster or under more general conditions.

One salient possibility is to devise some sort of weighted importance sampling version of our algorithm. Weighted importance sampling is widely known to produce lower variance estimates than conventional importance sampling, at the cost of introducing transient bias (bias that decreases to zero as the number of samples increases to infinity). For example, in our earlier work with table-lookup approximations (Prepol, Sutton and Singh, 2000), we discussed an importance sampling estimate

\[ Q^I(s, a) = \frac{\sum R_{\text{est}}}{N} \]

where each is a return in an episode under the behavior policy after an occurrence of state-action pair , and the weight is a product of importance sampling correction ratios \( \beta_i \), \( \beta_i = \frac{\pi_i(s) \pi'_i(s)}{\pi'_i(s)} \), where is the time of occurrence of , and the last time, within the episode). As is the current paper, this weight is chosen such that the product has the proper expected value for the target policy, i.e., such that \( R_i(s) = Q^T(s, a) \). By the law of large numbers, \( Q^I_N \) converges w.p.1 to \( Q^T(s, a) \) if the are bounded. But the might have infinite variance, and so \( Q^I_N \) might also have unbounded variance. However, the corresponding weighted version of this tabular importance sampling estimator,

\[ Q^IW_N(s, a) = \sum_{N_{\text{est}}} \frac{R_{\text{est}}}{N} \]

which also converges to \( Q^T \) w.p.1, has variance which goes to zero as grows, as we now show. First we need an additional definition and some standard results (e.g., Durrett, 1995).

Definition 1 A sequence of \( c_N \) converges in probability to \( H \), for any \( \epsilon > 0 \),

\[ \lim_{N \to \infty} P(|c_N - H| > \epsilon) = 0. \]

Theorem 5 (Weak Law of Large Numbers).

Let \( \{X_i\} \) be a sequence of i.i.d. random variables such that \( E[X_i] < \infty \). Then the estimator \( c_N = \frac{1}{N} \sum_{i=1}^{N} X_i \), converges in probability to \( E[X] \).

Using these results we can show:

Theorem 7 For \( \gamma < 1 \), \( \var(Q^IF_{N}) \) goes to zero as \( N \) goes to infinity.

Proof: First we show convergence in probability. We can write the estimator as a "top" part over a "bottom" part (dropping the everywhere):

\[ Q^IF_{N} = \frac{T_N}{B_N} \]
In the backward view, the eligibility trace at time $t$ is:

$$\delta_t = \sum_{k=0}^{T-1} \alpha \phi_k (\gamma \lambda)^k \prod_{j=0}^{k-1} R_j,$$

and the sum of the updates that occur during an episode is:

$$\sum_{k=0}^{T-1} \alpha \phi_k = \sum_{k=0}^{T-1} \alpha \phi_k \sum_{k=0}^{T-1} \phi_k (\gamma \lambda)^k \prod_{j=0}^{k-1} R_j = \sum_{k=0}^{T-1} \alpha \phi_k (R_0 - \delta^2 \phi_k).$$

5. An Empirical Illustration

To illustrate our algorithm we use the 11 x 11 grid-world environment depicted in Figure 2. The MDP is deterministic and has 4 actions, moving up, down, left or right. If the agent bumps into a wall, it remains in the same state. The four corner states are terminal. The agent receives a reward of +1 for the actions entering the bottom-right and upper-left corners, and −1 for entering the other two corners. All the other rewards are 0. The initial state is in the center, and the initial action is chosen randomly to be right or left. The target policy chooses down 40% of the time and up 10% of the time, with right and left chosen 25% of the time. The behavior policy is similar except with reversed up/down probabilities; it chooses down 10% of the time and up 40% of the time. In order to ensure that all the conditions of our convergence theorem are satisfied, trials are limited to 1000 time steps. However, this upper limit was never reached during our experiments.

The features used by the function approximator are overlapping stripes of width 3, parallel to the vertical axis. There are 13 such stripes. One consequence is that under the target policy, all actions from the leftmost column have negative value, whereas all actions from the rightmost column have negative values. The situation is reversed under the behavior policy. We implemented the incremental (backward view) version of importance sampled TD($\lambda$), with $\lambda = 0$ and $\lambda = 0.9$, and updating taking place only at the end of an episode. Because the results are very similar, we only present the data for $\lambda = 0.9$. The initial parameter of the function approximator was $\theta_0 = 0$.

6. On the Possibility of Weighted Importance Sampling Methods

We have introduced a new, off-policy version of linear TD($\lambda$) and shown that it converges near the best solution consistent with its structure. However, excessive variance remains an issue, and there may be algorithms that achieve the same asymptotic solution faster or under more general conditions.

One salient possibility is to devise some sort of weighted importance sampling version of our algorithm. Weighted importance sampling is widely known to produce lower variance estimates than conventional importance sampling, at the cost of introducing transient bias (bias that decreases to zero as the number of samples increases to infinity).

For example, in our earlier work with table-lookup approximations (Preuss, Sutton and Singh, 2000), we discussed an importance sampling estimate $Q^I(s, a) = \sum_{R_t} \frac{R_t}{N}$, where each $R_t$ is a return in an episode under the behavior policy after an occurrence of state-action pair $s, a$, and the weight $w_t$ is a product of importance sampling correction ratios $\rho_{s, a} \cdot \rho_{s', a'} \cdot \cdots \cdot \rho_{s, a}$ (where $t$ is the time of occurrence of $s, a$, and $T$ the last time, within the episode). As is the current paper, this weight is chosen such that the product $R_t w_t$ has the proper expected value for the target policy, i.e., such that $E_s\left(R_t w_t \right) = Q^T(s, a)$. By the law of large numbers, $Q^I$ converges w.p.1 to $Q^T(s, a)$ if the $R_t$ are bounded. But the $w_t$ might have infinite variance, and so $Q^I$ might also have unbounded variance. However, the corresponding weighted version of this tabular importance sampling estimator,

$$Q^{IF}(s, a) = \frac{\sum_{n=0}^{N} R_{n+1}}{\sum_{n=0}^{N} w_{n+1}},$$

which also converges to $Q^T(s, a)$ w.p.1, has variance which goes to zero as $N$ grows, as we now show. First we need an additional definition and some standard results (e.g. Duranton, 1990).

Definition 1 A sequence of $\epsilon_N$ converges in probability to $\epsilon$, if, for any $\epsilon > 0$, 

$$\lim_{N\to \infty} P_{\epsilon_N-H}(\epsilon_N-H > \epsilon) = 0.$$

Theorem 5 (Weak Law of Large Numbers). Let $\{X_n\}_{n=0}^\infty$ be a sequence of i.i.d. random variables such that $E[X_1] < \infty$. Then the estimator $\epsilon_N = \frac{1}{N} \sum_{n=1}^{N} X_n$ converges in probability to $E[X]$.

(Under some assumptions, the stronger law of large numbers (convergence w.p.1) also holds, but we will not need it for our result.)

Theorem 6 If $|\epsilon_N|$ is bounded, then convergence in probability of $\epsilon_N$ to $\epsilon$ implies that $\lim_{N\to \infty} \text{var}(\epsilon_N) = 0$.

Proof: Suppose that $|\epsilon_N| \leq C$, for some constant $C$. This also implies that $|\epsilon_N-H| \leq C$. Pick any $\epsilon > 0$. Then:

$$\text{var}(\epsilon_N) \leq E[|\epsilon_N-H|^2] \leq c^2 + 4C^2 \cdot P_{\epsilon_N-H}(\epsilon_N-H > \epsilon).$$

Now take the limit as $N \to \infty$. Since the $\epsilon_N$ converges in probability, the rightmost term goes to zero and so $\lim_{N\to \infty} \text{var}(\epsilon_N) \leq c^2$.

But this is true for any $\epsilon > 0$, so the theorem follows.

Using these results we can show:

Theorem 7 For $\gamma < 1$, $\text{var}(Q^IF)$ goes to zero as $N$ goes to infinity.

Proof: First we show convergence in probability. We can write the estimator as a "top" part over a "bottom" part (dropping the $s, a$ everywhere):

$$Q^{IF} = \frac{T_n}{B_n}.$$
Machine Learning: Proceedings of the Eighteenth International Conference

where

\[ T_N = \frac{1}{N} \sum_{i=1}^{N} R_i w_i \]

and

\[ B_N = \frac{1}{N} \sum_{i=1}^{N} B_i w_i. \]

Because \( E(w_i) = 1 \) is finite, we can apply the weak law of large numbers twice here to show that \( T_N \) converges in probability to \( Q^* \) and \( B_N \) converges in probability to \( 1. \) Thus we know that the top is very close to \( Q^* \) except for a tiny probability, and the bottom is very close to \( 1 \) except for a tiny probability. Now we can ignore what happens with tiny probability; that will correspond to the tiny probability with which \( Q^{SW} \) is allowed to be significantly different from \( Q^* \). So consider the cases when top and bottom are very near \( Q^* \) and 1 respectively. If we pick the "very near" close enough, then we can also bound the difference of

\[ Q^S \text{ from } Q^* \. \]

So we get that \( Q^{SW} \) is arbitrarily close to \( Q^* \) except for an arbitrarily small probability, i.e., \( Q^{SW} \) converges in probability to \( Q^* \).

Now we seek to apply Theorem 6, for which we need only to show that \( Q^{SW} \) is bounded. From its definition, \( Q^{SW} \) can clearly be no greater than the largest possible \( |R_i| \). For bounded individual rewards and \( \gamma < 1 \), we have \( |R_i| < \infty \). Thus, Theorem 6 applies and so \( \lim_{N \to \infty} \varphi(Q^{SW}) = 0.0 \).

Thus, in the tabular case the weighted algorithm vanishing variance. The same cannot be said for the conventional importance sampling algorithm. It seems plausible that a similar pattern of results could hold for the case with linear function approximation. To explore this possibility would of course require some form of weighted importance sampling that was consistent with function approximation.

Acknowledgements

The authors gratefully thank Csaba Szepesvari for pointing out a problem in the previous version of our convergence proof, and David McAllester and Salinder Singh for many helpful discussions about importance sampling.

References


Multiple Instance Regression

Soumya Ray
BRAY@CS.WISC.EDU
Department of Computer Sciences and Department of Biostatistics and Medical Informatics, University of Wisconsin, Madison, WI 53706

David Page
PAGIR@BIOSTAT.WISC.EDU
Department of Biostatistics and Medical Informatics and Department of Computer Sciences, University of Wisconsin, Madison, WI 53706

Abstract

This paper introduces multiple instance regression, a variant of multiple regression in which each data point may be described by more than one vector of values for the independent variables. The goals of this work are to (1) understand the computational complexity of the multiple instance regression task and (2) develop an efficient algorithm that is superior to ordinary multiple regression when applied to multiple instance data sets.

1. Introduction

The multiple instance problem (Dieterich et al., 1997) arises when the classification of every data point is not known uniquely. For instance, we might know that one of the attribute vectors \( X_1, X_2, X_3 \), or both, are responsible for a data point being classified as belonging to a certain class, but we may be unable to pinpoint which vector. This is frequently the case. For example, in drug design, we wish to distinguish molecules effective as drugs from ineffective ones. Here, training examples are in the form of conformations (3D structures) of a molecule, along with its class (active/inactive). However, a molecule may exist in a dynamic equilibrium of several conformations. While the observed activity will be a function of one or more of these conformations, it is typically impossible to determine which one(s). On the other hand, it is almost never the case that all conformations contribute to the observed activity. Hence it is desirable to learn a classifier which can take the multiple instance nature of these examples into account. Multiple instance problems arise in a variety of other domains as well, ranging from in vitro fertilization (Saith et al., 1997) to image analysis (Maros & Lezcano-Pérez, 1998).

It is worthwhile to note that in several applications of the multiple instance problem, the actual predictions desired are real valued. The drug design example is a case in point. While it is beneficial to be able to predict the active or inactive classification, our experience is that drug developers often prefer to see predicted activity levels of these molecules, expressed as real numbers. Most past research on the multiple instance problem has focused on the design of discrete classifiers. We investigate instead the task of learning to predict the value of a real valued dependent variable, under the assumptions of multiple regression, for data where the multiple instance problem is present. We call this task multiple instance multiple regression, or for brevity multiple instance regression.

Our investigation of multiple instance regression has two goals. The first is to understand the computational complexity inherent in the task of multiple instance regression—for example, we would like to know if a linear time algorithm exists as for ordinary regression. The second goal is to determine whether multiple instance regression has any advantage over ordinary regression when building classifiers for data sets where the multiple instance problem is present. In such cases, we could simply ignore the multiple instance problem, treating each instance as a distinct data point having the classification of the bag, and use ordinary regression. This is effectively the approach taken by Sinivuosa and Cosmeño (1999) to incorporate linear regression literals into inductive logic programming (see Section 6). We wish to understand if multiple instance regression confers any benefit over this baseline method.

2. Task Definition

We define the task under consideration as follows. We are given a set of n bags. The ith bag consists of \( m_i \) instances and a real valued class label \( y_i \). Instance \( j \)