ON SAMPLING ERROR IN BATCH ACTION-VALUE PREDICTION ALGORITHMS

Brahma S. Pavse[1,2]*, Josiah P. Hanna[3], Ishan Durugkar[2], Peter Stone[2,4]
[1]Salesforce, [2]University of Texas at Austin, [3]University of Edinburgh, [4]Sony AI
{brahmasp,ishand,pstone}@cs.utexas.edu
josiah.hanna@ed.ac.uk

ABSTRACT

Estimating a policy's action-values is a fundamental aspect of reinforcement learning. In this work, we study the application of TD methods for learning action-values in an offline setting with a fixed batch of data. Motivated by recent work (Pavse et al., 2020), we observe that a fixed batch of offline data may contain two forms of distribution shift: the data may be collected from a different behavior policy than the target policy (off-policy data) and the empirical distribution of the data may differ from the sampling distribution of the data (sampling error). In this work, we focus on the second problem by analyzing the sampling error that arises due to variance in sampling from a finite-sized batch of data in the *on-policy offline* RL setting. We study how action-value learning algorithms suffer from this *sampling error* by considering their so-called *certainty-equivalence estimates* (Sutton, 1988; Pavse et al., 2020). We prove that each algorithm uses its certainty-equivalence estimates of the policy and transition dynamics to converge to its respective fixed-point. We then empirically evaluate each algorithm's performance by measuring the mean-squared value error on Gridworld. Ultimately, we find that by reducing sampling error, an algorithm can produce significantly accurate action-value estimations.

1 INTRODUCTION

Reinforcement learning (RL) (Sutton & Barto, 2018) algorithms have become prevalent in sequential-decision making problems such as robot manipulation (Kober et al., 2013; Gu et al., 2016) and autonomous driving (Sallab et al., 2017). Many RL algorithms learn optimal control by computing the *action-values*, a function that estimates the expected return when starting in a state, taking a particular action, and thereafter following a particular policy (Mnih et al., 2013; Rummery & Niranjan, 1994a; van Seijen et al., 2009; Watkins & Dayan, 1992). In this study, we consider three action-value prediction algorithms and study them from the novel perspective of *policy sampling error* (Pavse et al., 2020). According to Pavse et al. (2020), policy sampling error occurs when a batch algorithm makes learning updates weighted according to the frequency of an action appearing in the batch instead of the true frequency of that action.

Sutton (1988) and Pavse et al. (2020) showed that batch TD(0) converges to the so-called *certainty-equivalence estimate* (CEE), which is the value function learned using the maximum likelihood estimates (MLE) of the policy and transition dynamics according to the batch of data. In this work, we prove that batch action-value TD, the prediction variant of SARSA (Rummery & Niranjan, 1994a), converges to a version of the CEE that gives action-values for the empirical policy *instead* of the desired policy. Pavse et al. (2020) call this mismatch in policy distribution *policy sampling error*. We also prove that batch expected action-value TD, a prediction algorithm based on Expected SARSA (van Seijen et al., 2009), eliminates this policy sampling error by taking an expectation over the desired policy on the bootstrapped state. Finally, we also introduce batch PSEC action-value TD, a novel prediction algorithm based on PSEC-TD(0) (Pavse et al., 2020), which uses importance sampling (Precup et al., 2001) to eliminate the sampling error by correcting learning from the empirical policy to the true desired policy. The contributions of the paper are as follows:

*Work done while at UT Austin

1. Derive the fixed points of the above algorithms under a fixed batch of data.
2. Analyze the sampling error of the algorithms from the derived fixed-points.
3. Empirically validate our analysis by comparing the mean-squared value error (MSVE) of the different algorithms.

A difficulty when learning in the offline RL setting is that there are distribution mismatches due to: 1) difference in the target policy and behavior policy and 2) variance in sampling from the finite batch of data. In this paper, we isolate the latter problem by considering the offline on-policy setting. Understanding this problem serves as a first step towards understanding the sampling error in the more general off-policy case.

2 Background

This section introduces notation and formally specifies the batch action-value learning problem.

2.1 Notation and Definitions

We consider a Markov decision process (MDP) with state space $\mathcal{S}$, action space $\mathcal{A}$, reward function, R, transition dynamics function P, and discount factor γ (Puterman, 2014). In any state, s, an agent selects stochastic actions according to a policy π, $a \sim \pi(\cdot|s)$. After taking an action, a, in state s the agent transitions to a new state $s' \sim P(\cdot|s, a)$ and receives reward $R(s, a, s')$. We assume $\mathcal{S}$ and $\mathcal{A}$ to be finite. We consider the episodic, discounted, and finite horizon setting. We are concerned with computing the action-value function, $q^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. The action-value function estimates the expected discounted return when starting in a state, taking a particular action, and thereafter following a particular policy:

$$q^\pi(s, a) := \mathbf{E}_\pi\left[\sum_{k=0}^{L} \gamma^k R_{t+k+1} \,\middle|\, S_t = s, A_t = a\right], \forall s \in \mathcal{S}, \forall a \in \mathcal{A}$$

where L is the terminal time-step and the expectation is taken over the distribution of future states, actions, and rewards under π and P. We analyze action-value function learning in the tabular setting.

2.2 Batch Action-Value Prediction

This work investigates the problem of approximating, q^π, given a batch of data, $\mathcal{D}$, and a policy, π, i.e. *policy evaluation*. Let a single episode, τ, be defined as $\tau := (S_0, A_0, R_0, S_1, ..., S_{L_{\tau-1}}, A_{L_{\tau-1}}, R_{L_{\tau-1}})$, where L_τ is the length of the episode τ. The batch of data, $\mathcal{D}$, consists of m episodes, i.e., $\mathcal{D} := \{\tau_i\}_{i=0}^{m-1}$. Our focus is in the offline *on-policy* setting, where $\mathcal{D}$ is generated by π.

In batch action-value prediction, an action-value function learning algorithm uses a fixed batch of data to learn an estimate, $\widehat{q}^\pi$, that approximates the true action-value function, q^π.

The error of the predicted value function, $\widehat{q}^\pi$, with respect to the true value function, q^π, is measured by calculating the mean squared value error between $q^\pi(s, a)$ and $\widehat{q}^\pi(s, a)\ \forall s \in \mathcal{S}, \forall a \in \mathcal{A}$ weighted by the proportion of time spent in each state-action pair under policy π, $d_\pi(s, a)$. Thus, we seek to find $\widehat{q}^\pi$ that minimizes:

$$\sum_{s \in \mathcal{S}, a \in \mathcal{A}} d_\pi(s, a)\Big(q^\pi(s, a) - \widehat{q}^\pi(s, a)\Big)^2 \tag{1}$$

3 Convergence of Batch Action-Value Prediction Algorithms

In this section, we do the following: 1) introduce the three batch action-value learning algorithms, 2) give theorems stating the fixed-point that each of these algorithms converge to, and 3) relate these to the certainty-equivalence (Sutton, 1988; Pavse et al., 2020) and give remarks on their policy sampling errors when estimating the action-value function.

First, we introduce additional notation. Let $\widehat{\mathcal{S}}$ be the set of states and $\widehat{\mathcal{A}}$ be the set of actions that appear in $\mathcal{D}$ and let $\bar{R}(s,a)$ be the mean reward observed in state s upon taking action a in the batch $\mathcal{D}$. If the notation includes a hat ($\hat{}$), it is the maximum-likelihood estimate (MLE) according to $\mathcal{D}$. We also refer to the MLE policy as the *empirical* policy. For example, $\hat{P}$ is the MLE of P. Let $\hat{c}(s)$ be the number of occurrences of s in $\mathcal{D}$. $\beta_n(s,a) = \alpha\hat{c}(s)$ is a suitable step-size according to the stochastic approximation conditions (see Appendix A). The certainty-equivalence estimates of action-values are the estimates learned using the empirical policy and MLE transition dynamics according to the batch of data.

We note that batch action-value TD and batch expected action-value TD are similar to SARSA (Rummery & Niranjan, 1994a) and Expected SARSA (van Seijen et al., 2009) respectively with the exception that the former pair are batch algorithms and are policy evaluation algorithms i.e. they evaluate action-values for a *fixed* policy. In batch action-value prediction, the algorithms first accumulate all their respective errors for each transition in the batch and then make an aggregated update to the action-value. Due to space constraints, we include Algorithm 1 in Appendix B, which gives pseudo-code for the batch action-value TD and batch expected action-value TD.

3.1 Batch Action-Value TD

For batch action-value TD, we have the following batch update rule when processing the batch for the $n+1^{\text{th}}$ iteration:

$$q(s,a)_{n+1} \leftarrow q(s,a)_n + \beta_n(s,a) \sum_{s' \in \widehat{\mathcal{S}}} \sum_{a' \in \widehat{\mathcal{A}}} \widehat{P}(s'|s,a)\hat{\pi}(a'|s,a,s') \left[\bar{R}(s,a) + \gamma q(s',a')_n - q(s,a)_n\right] \tag{2}$$

Given the certainty-equivalence estimates of the policy and transition dynamics, we then have the following fixed-point given by the Bellman equation:

$$q_{\text{AV-TD}}(s,a) = \bar{R}(s,a) + \gamma \sum_{s' \in \widehat{\mathcal{S}}} \widehat{P}(s'|s,a) \sum_{a' \in \widehat{\mathcal{A}}} \hat{\pi}(a'|s,a,s') q_{\text{AV-TD}}(s',a') \tag{3}$$

Given Equation (2) and (3), we have the following theorem (refer to Appendix A for proof details):

Theorem 1 (Batch Action-Value TD Convergence)**.** *Batch action-value TD, as defined by update (2), converges to the action-value function given by (3) when the following assumptions hold:*

1. *$\mathcal{S}$ and $\mathcal{A}$ are finite.*
2. *$\sum_n \beta_n(s,a) = \infty$, $\sum_n (\beta_n(s,a))^2 < \infty$.*
3. *The reward function is bounded.*

From Theorem 1, for a finite batch of data, batch action-value TD computes the action-value function for empirical policy and MLE transition dynamics instead of the true policy and transition dynamics, resulting in policy *and* transition dynamics sampling error.

3.2 Batch Expected Action-Value TD

Taking the same approach as above, we have the batch update for batch expected action-value TD when processing the batch for the $n+1^{\text{th}}$ iteration:

$$q(s,a)_{n+1} \leftarrow q(s,a)_n + \beta_n(s,a) \sum_{s' \in \widehat{\mathcal{S}}} \hat{P}(s'|s,a) \left[\bar{R}(s,a) + \gamma \sum_{a' \in \mathcal{A}} \pi(a'|s') q(s',a')_n - q(s,a)_n\right] \tag{4}$$

Given the certainty-equivalence estimates of the transition dynamics, we then have the fixed-point given by the Bellman equation:

$$q_{\text{EXP-AV-TD}}(s,a) = \bar{R}(s,a) + \gamma \sum_{s' \in \widehat{\mathcal{S}}} \widehat{P}(s'|s,a) \sum_{a' \in \mathcal{A}} \pi(a'|s') q_{\text{EXP-AV-TD}}(s',a') \tag{5}$$

Given Equations (4) and (5), we have the following theorem (refer to Appendix A for proof details):

Theorem 2 (Batch expected action-value TD Convergence)**.** *Batch expected action-value TD(0), as defined by update (4), converges to the action-value function given by (5) when the following assumptions hold:*

1. *$\mathcal{S}$ and $\mathcal{A}$ are finite.*
2. *$\sum_n \beta_n(s,a) = \infty$, $\sum_n (\beta_n(s,a))^2 < \infty$.*
3. *The reward function is bounded.*

From Theorem 2, we see that batch expected action-value TD computes the action-value function for the MLE transition dynamics instead of the true dynamics. However, it computes a more accurate action-value than batch action-value TD since by computing the action-value for the *true* desired policy the former reduces sampling error.

3.3 Batch PSEC Action-Value TD

PSEC-TD(0) (Pavse et al., 2020) was introduced as an algorithm to correct for the same policy sampling error in batch TD(0). Similarly, we view the convergence of batch action-value TD as an off-policy problem since it evaluates action-values for the *empirical* instead of true policy. As introduced by Pavse et al. (2020), this type of problem can be corrected for by estimating the empirical policy and then using importance sampling to correct learning from the empirical policy to the true policy. We call the policy sampling error corrected batch action-value TD algorithm as batch PSEC action-value TD. We have the batch PSEC action-value TD update on the $n+1$ iteration of processing the batch as follows:

$$q(s,a)_{n+1} \leftarrow q(s,a)_n + \beta_n(s,a) \sum_{s' \in \widehat{\mathcal{S}}} \widehat{P}(s'|s,a) \sum_{a' \in \widehat{\mathcal{A}}} \frac{\pi(a'|s')}{\hat{\pi}(a'|s,a,s')} \hat{\pi}(a'|s,a,s') \left[\bar{R}(s,a) + \gamma q(s',a')_n - q(s,a)_n \right] \tag{6}$$

where $\frac{\pi(a'|s')}{\hat{\pi}(a'|s,a,s')}$ is the PSEC weight to correct learning from $\hat{\pi}$ to π. Given the certainty-equivalence estimates of the transition dynamics, we then have the fixed-point given by the Bellman equation:

$$q(s,a)_{\text{PSEC-AV-TD}} = \bar{R}(s,a) + \gamma \sum_{s' \in \widehat{\mathcal{S}}} \widehat{P}(s'|s,a) \sum_{a' \in \widehat{\mathcal{A}}} \pi(a'|s') q_{\text{PSEC-AV-TD}}(s',a') \tag{7}$$

Given Equations (6) and (7), we have the following theorem (refer to Appendix A for proof details):

Theorem 3 (Batch PSEC action-value TD Convergence)**.** *Batch PSEC action-value TD, as defined by update (6), converges to the action-value function given by (7) when the following assumptions hold:*

1. *$\mathcal{S}$ and $\mathcal{A}$ are finite.*
2. *$\sum_n \beta_n(s,a) = \infty$, $\sum_n (\beta_n(s,a))^2 < \infty$.*
3. *The reward function is bounded.*

From Theorem 3, we see that batch PSEC action-value TD is similar to batch expected action-value TD in that it also computes the action-value function for the MLE transition dynamics and *true* desired policy. However, Equation (7) computes the expectation over available actions in the batch, whereas Equation (4) computes the expectation over all possible actions. Thus, batch PSEC action-value TD converges to same certainty-equivalence estimate as batch expected action-value TD under an action visitation condition.

3.4 Remarks on Policy Sampling Error

From Theorems 1, 2, and 3, we see that batch expected action-value TD and batch PSEC action-value TD correct for the policy sampling error while batch action-value TD does not since the former two evaluate action-values for the true policy. This correction is also reflected in our empirical results (Section 4). Between the two, we find that batch expected action-value TD is better suited for action-value learning than batch PSEC action-value TD since: 1) it achieves better performance with less data (Section 4) and 2) it does not incur the MLE computational cost, which can be $O(|\mathcal{S}||\mathcal{A}|)$, where $|\mathcal{S}|$ and $|\mathcal{A}|$ are the number of states and actions respectively.

4 Empirical Results

In this section, we empirically study the performance of the batch action-value prediction algorithms as a function of data size.

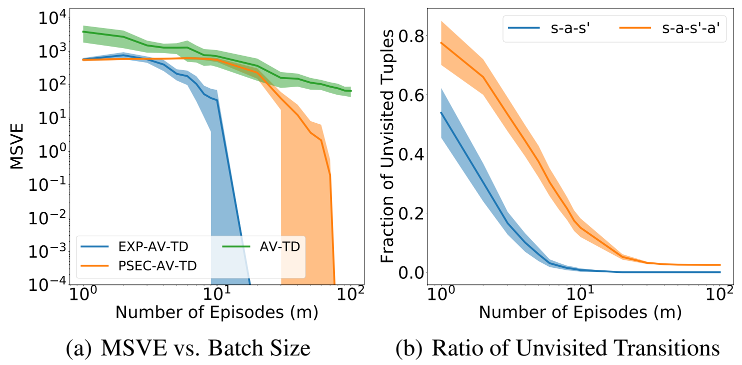

Figure 1: Deterministic Gridworld. Both axes of Figure 1(a) and x-axis of Figure 1(b) are log-scaled. Figure 1(a) shows errors against the true action-values. Figure 1(b) shows ratio of unvisited transition tuples as a function of data size. All metrics are computed over 30 trials with 95% confidence intervals.

Figure 1(a) illustrates the performance of the three algorithms against the true action-values on a deterministic gridworld. We see that for a given batch size, batch expected action-value TD and batch PSEC action-value TD perform better than batch action-value TD. We find that batch expected action-value TD and batch PSEC action-value TD improve in performance as more (s, a, s') and (s, a, s', a') tuples are visited respectively. This finding aligns with Figure 1(b), which shows the unvisited tuple ratio as a function of batch size. As the unvisited ratio of both these types of tuples goes to 0, both algorithms correct for more policy sampling error and converge to the same certainty-equivalence (Equations (5) and (7)). However, we can see that batch PSEC action-value TD requires more data to converge to the same CEE as batch expected action-value TD. Note that in *deterministic* Gridworld, there is no transition dynamics sampling error; thus, these algorithms are able to achieve 0 MSVE.

5 Related Work

In this section, we discuss the related literature.

Many works avoid policy sampling error by performing analytic computations. Expected SARSA (van Seijen et al., 2009) was introduced as a lower variance algorithm to SARSA (Rummery & Niranjan, 1994a) in order to achieve faster learning. The Tree-backup algorithm (Precup et al., 2000) extends Expected SARSA to a multi-step algorithm. $Q(\sigma)$ (Asis et al., 2017) unifies SARSA (Sutton, 1996; Rummery & Niranjan, 1994b), Expected SARSA, and Tree-backups, to find a balance between sampling and analytic expectation computation. Our work is distinct from these in the following ways: 1) we focus on policy evaluation of the action-values for a fixed policy and 2) we study the relation between the policy sampling error in these action-value prediction algorithms and their certainty-equivalence estimate.

Batch PSEC action-value TD is an extension to PSEC-TD(0) (Pavse et al., 2020), an algorithm for batch *state-value* learning. Batch PSEC action-value TD's technique of estimating the policy from the batch and then using importance sampling has also been well-studied for policy evaluation (Hanna et al., 2019), policy gradient learning (Hanna & Stone, 2019), multi-armed bandits (Li et al., 2015; Narita et al., 2018; Xie et al., 2018), causal inference (Hirano et al., 2003; Rosenbaum, 1987), and

Monte Carlo integration (Henmi et al., 2007; Delyon & Portier, 2016). To the best of our knowledge, this technique has *not* been studied for on-policy action-value prediction.

6 Summary

In this work, we isolated one type of distribution shift that arises in the offline RL setting where the empirical distribution of data may differ from the sampling distribution of the data by focusing on the *on-policy offline* RL setting. We studied three batch action-value prediction algorithms from the novel perspective of understanding their policy sampling error and relating them to the certainty-equivalence estimate (CEE). We proved that batch action-value TD evaluates the empirical policy instead of the desired true policy. We then proved that batch expected action-value TD and batch PSEC action-value TD, on the other hand, compute the action-values for the desired policy. We proved that batch expected action-value TD and batch PSEC action-value TD converge to the same CEE (under a transition tuple visitation condition). We reached these conclusions through convergence proofs and an empirical analysis. An important line of future work will be to incorporate the second source of distribution mismatch between the behavior policy and target policy i.e. the off-policy setting.

Acknowledgements

This work has taken place in the Learning Agents Research Group (LARG) at UT Austin. LARG research is supported in part by NSF (CPS-1739964, IIS-1724157, NRI-1925082), ONR (N00014-18-2243), FLI (RFP2-000), ARO (W911NF-19-2-0333), DARPA, Lockheed Martin, GM, and Bosch. Peter Stone serves as the Executive Director of Sony AI America and receives financial compensation for this work. The terms of this arrangement have been reviewed and approved by the University of Texas at Austin in accordance with its policy on objectivity in research.

References

Kristopher De Asis, J. Fernando Hernandez-Garcia, G. Zacharias Holland, and Richard S. Sutton. Multi-step reinforcement learning: A unifying algorithm, 2017.

Bernard Delyon and Franois Portier. Integral approximation by kernel smoothing. *Bernoulli*, 22 (4):2177?2208, Nov 2016. ISSN 1350-7265. doi: 10.3150/15-bej725. URL http://dx.doi.org/10.3150/15-BEJ725.

Shixiang Gu, Ethan Holly, Timothy P. Lillicrap, and Sergey Levine. Deep reinforcement learning for robotic manipulation. *CoRR*, abs/1610.00633, 2016. URL http://arxiv.org/abs/1610.00633.

Josiah Hanna and Peter Stone. Reducing sampling error in the monte carlo policy gradient estimator. In *Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, May 2019.

Josiah Hanna, Scott Niekum, and Peter Stone. Importance sampling policy evaluation with an estimated behavior policy. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, June 2019.

Masayuki Henmi, Ryo Yoshida, and Shinto Eguchi. Importance sampling via the estimated sampler. *Biometrika*, 94(4):985–991, 2007. URL https://EconPapers.repec.org/RePEc:oup:biomet:v:94:y:2007:i:4:p:985-991.

Keisuke Hirano, Guido W. Imbens, and Geert Ridder. Efficient estimation of average treatment effects using the estimated propensity score. *Econometrica*, 71(4):1161–1189, 2003. doi: 10.1111/1468-0262.00442. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/1468-0262.00442.

Tommi Jaakkola, Michael I. Jordan, and Satinder P. Singh. Convergence of stochastic iterative dynamic programming algorithms. In *Proceedings of the 6th International Conference on Neural Information Processing Systems*, NIPS93, pp. 703710, San Francisco, CA, USA, 1993. Morgan Kaufmann Publishers Inc.

Jens Kober, J. Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32:1238–1274, 09 2013. doi: 10.1177/0278364913495721.

Lihong Li, Rémi Munos, and Csaba Szepesvári. Toward minimax off-policy value estimation. In *AISTATS*, 2015.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning, 2013.

Yusuke Narita, Shota Yasui, and Kohei Yata. Efficient counterfactual learning from bandit feedback. *CoRR*, abs/1809.03084, 2018. URL `http://arxiv.org/abs/1809.03084`.

Brahma Pavse, Ishan Durugkar, Josiah Hanna, and Peter Stone. Reducing sampling error in batch temporal difference learning. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, July 2020.

Doina Precup, Richard S. Sutton, and Satinder P. Singh. Eligibility traces for off-policy policy evaluation. In *ICML*, 2000.

Doina Precup, Richard S. Sutton, and Sanjoy Dasgupta. Off-policy temporal difference learning with function approximation. In *ICML*, 2001.

Martin L. Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Paul R. Rosenbaum. Model-based direct adjustment. *Journal of the American Statistical Association*, 82(398):387–394, 1987. ISSN 01621459. URL `http://www.jstor.org/stable/2289440`.

G. Rummery and Mahesan Niranjan. On-line q-learning using connectionist systems. *Technical Report CUED/F-INFENG/TR 166*, 11 1994a.

G. A. Rummery and M. Niranjan. On-line q-learning using connectionist systems. Technical report, 1994b.

Ahmad EL Sallab, Mohammed Abdou, Etienne Perot, and Senthil Yogamani. Deep reinforcement learning framework for autonomous driving. *Electronic Imaging*, 2017(19):70–76, 2017.

Satinder Singh, Tommi Jaakkola, Michael Littman, and Csaba Szepesvri. Convergence results for single-step on-policy reinforcement-learning algorithms. *Machine Learning*, 38:287–308, 03 2000. doi: 10.1023/A:1007678930559.

Richard S Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 3 (1):9–44, 1988.

Richard S Sutton. Generalization in reinforcement learning: Successful examples using sparse coarse coding. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo (eds.), *Advances in Neural Information Processing Systems 8*, pp. 1038–1044. MIT Press, 1996.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018. URL `http://incompleteideas.net/book/the-book-2nd.html`.

H. van Seijen, H. van Hasselt, S. Whiteson, and M. Wiering. A theoretical and empirical analysis of expected sarsa. In *2009 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning*, pp. 177–184, March 2009. doi: 10.1109/ADPRL.2009.4927542.

Christopher J. C. H. Watkins and Peter Dayan. Q-learning. In *Machine Learning*, pp. 279–292, 1992.

Yuan Xie, Boyi Liu, Qiang Liu, Zhaoran Wang, Yuan Zhou, and Jian Peng. Off-policy evaluation and learning from logged bandit feedback: Error reduction via surrogate policy. *CoRR*, abs/1808.00232, 2018. URL `http://arxiv.org/abs/1808.00232`.

A Convergence Proofs

In proving Theorems 1, 2, and 3, we used Lemma 1 from Singh et al. (2000) (also used by van Seijen et al. (2009); Jaakkola et al. (1993)), which states the following:

Lemma 4. *Consider a stochastic process (ζ_t, Δ_t, F_t), where $\zeta_t, \Delta_t, F_t : X \to \mathbb{R}$ satisfies the equation:*

$$\Delta_{t+1}(x) = (1 - \zeta_t(x))\Delta_t(x) + \zeta_t(x)F_t(x)$$

where $x \in X$ and $t = 0, 1, 2....$ Assume that the following hold:

1. *the set X is finite.*
2. *$\zeta_t(x) \in [0, 1]$ $\sum_t \zeta_t(x) = \infty$, $\zeta_t^2(x) < \infty$, w.p. 1.*
3. *$||\mathbf{E}[F_t|P_t]|| \leq \kappa||\Delta_t|| + c_t$, where $\kappa \in [0, 1)$ and c_t converges to 0 w.p. 1.*
4. *$Var[F_t(x)|P_t] \leq K(1 + \kappa||\Delta_t||^2)$, for some constant, K.*

Here $P_t = \{\Delta_t, \Delta_{t-1}, ..., F_{n-1}, ..., \zeta_{t-1}...\}$ stands for the past at step t. F_t and ζ_t are allowed to depend on the past insofar as the above conditions remain valid. $||.||$ denotes the maximum norm. Then Δ_t converges to 0 w.p. 1.

In our proofs below, we apply Lemma 4 with $X = \widehat{\mathcal{S}} \times \widehat{\mathcal{A}}$. We now include the full proof details for each theorem.

Theorem 1 (Batch Action-Value TD Convergence)**.** *Batch action-value TD, as defined by update (2), converges to the action-value function given by (3) when the following assumptions hold:*

1. *$\mathcal{S}$ and $\mathcal{A}$ are finite.*
2. *$\sum_n \beta_n(s, a) = \infty$, $\sum_n(\beta_n(s, a))^2 < \infty$.*
3. *The reward function is bounded.*

Proof. The proof strategy is as follows: we want to apply Lemma 4 to show convergence to Equation (3). In order to apply Lemma 4, the four conditions need to be met. The first and second conditions of Lemma 4 are met by the first two assumptions of the theorem. We discuss meeting the fourth condition at the end of the proof. For the third condition of Lemma 4, we must adapt Equation (2) so that is applicable. We define $\hat{c}(s, a, s', a')$ as the number of occurrences of transition tuple (s, a, s', a'). Notice that $\hat{c}(s, a, s', a') = \hat{c}(s)\hat{\pi}(a|s)\hat{P}(s'|s, a)\hat{\pi}(a'|s, a, s')$.

Consider the update rule of batch action-value TD for policy evaluation on the $n + 1$th iteration of processing the batch of data:

$$\begin{aligned}
q_{n+1}(s, a) &= q_n(s, a) + \alpha \sum_{s' \in \widehat{\mathcal{S}}} \sum_{a' \in \widehat{\mathcal{A}}} \hat{c}(s, a, s', a') \left[\bar{R}(s, a) + \gamma q_n(s', a') - q_n(s, a)\right] \\
&= q_n(s, a) + \alpha\hat{c}(s, a) \sum_{s' \in \widehat{\mathcal{S}}} \sum_{a' \in \widehat{\mathcal{A}}} \hat{P}(s'|s, a)\hat{\pi}(a'|s, a, s')[\bar{R}(s, a) + \gamma q_n(s', a') - q_n(s, a)] \\
&= q_n(s, a) + \beta_n(s, a) \sum_{s' \in \widehat{\mathcal{S}}} \sum_{a' \in \widehat{\mathcal{A}}} \hat{P}(s'|s, a)\hat{\pi}(a'|s, a, s')[\bar{R}(s, a) + \gamma q_n(s', a') - q_n(s, a)]
\end{aligned}$$

where $\beta_n(s, a) = \alpha\hat{c}(s, a)$. We now subtract $q_{\text{AV-TD}}(s, a)$ (Equation (3)) from both sides.

$$\begin{aligned}
q_{n+1}(s, a) - q_{\text{AV-TD}}(s, a) &= q_n(s, a) - q_{\text{AV-TD}}(s, a) \\
&\quad + \beta_n(s, a) \sum_{s' \in \widehat{\mathcal{S}}} \sum_{a' \in \widehat{\mathcal{A}}} \hat{P}(s'|s, a)\hat{\pi}(a'|s, a, s')[\bar{R}(s, a) + \gamma q_n(s', a') - q_n(s, a)]
\end{aligned}$$

Then adding and subtracting $\beta_n(s, a) \sum_{s' \in \widehat{\mathcal{S}}} \sum_{a' \in \widehat{\mathcal{A}}} \hat{P}(s'|s, a)\hat{\pi}(a'|s, a, s')q_{\text{AV-TD}}(s, a)$ on RHS. We also define $\Delta_n(s, a) = q_n(s, a) - q_{\text{AV-TD}}(s, a)$.

$$\begin{aligned}
\Delta_{n+1}(s,a) &= \Delta_n(s,a) + \beta_n(s,a) \sum_{s' \in \widehat{\mathcal{S}}} \sum_{a' \in \widehat{\mathcal{A}}} \hat{P}(s'|s,a)\hat{\pi}(a'|s,a,s')[\bar{R}(s,a) + \gamma q_n(s',a') - q_n(s,a)] \\
&\quad + \beta_n(s,a) \sum_{s' \in \widehat{\mathcal{S}}} \sum_{a' \in \widehat{\mathcal{A}}} \hat{P}(s'|s,a)\hat{\pi}(a'|s,a,s') q_{\text{AV-TD}}(s,a) - \beta_n(s,a) \sum_{s' \in \widehat{\mathcal{S}}} \sum_{a' \in \widehat{\mathcal{A}}} \hat{P}(s'|s,a)\hat{\pi}(a'|s,a,s') q_{\text{AV-TD}}(s,a) \\
&= (1 - \beta_n(s,a) \sum_{s' \in \widehat{\mathcal{S}}} \sum_{a' \in \widehat{\mathcal{A}}} \hat{P}(s'|s,a)\hat{\pi}(a'|s,a,s'))\Delta_n(s,a) \\
&\quad + \beta_n(s,a) \sum_{s' \in \widehat{\mathcal{S}}} \sum_{a' \in \widehat{\mathcal{A}}} \hat{P}(s'|s,a)\hat{\pi}(a'|s,a,s')[\bar{R}(s,a) + \gamma q_n(s',a') - q_{\text{AV-TD}}(s,a)] \\
\Delta_{n+1}(s,a) &= (1 - \beta_n(s,a))\Delta_n(s,a) + \beta_n(s,a) \sum_{s' \in \widehat{\mathcal{S}}} \sum_{a' \in \widehat{\mathcal{A}}} \hat{P}(s'|s,a)\hat{\pi}(a'|s,a,s')[\bar{R}(s,a) + \gamma q_n(s',a') - q_{\text{AV-TD}}(s,a)]
\end{aligned}$$

where $\sum_{s' \in \widehat{\mathcal{S}}} \sum_{a' \in \widehat{\mathcal{A}}} \hat{P}(s'|s,a)\hat{\pi}(a'|s,a,s') = 1$.

Then by comparing $\Delta_{n+1}(s,a)$ with Lemma 4, we have $\zeta_t = \beta_n$ and $F_t = \sum_{s' \in \widehat{\mathcal{S}}} \sum_{a' \in \widehat{\mathcal{A}}} \hat{P}(s'|s,a)\hat{\pi}(a'|s,a,s')[\bar{R}(s,a) + \gamma q_n(s',a') - q_{\text{AV-TD}}(s,a)]$. Then for the third condition of Lemma 4,

$$\begin{aligned}
||\mathbf{E}[F_t|P_t]|| &= ||\mathbf{E}[\bar{R}(s,a) + \gamma q(s',a') - q_{\text{AV-TD}}(s,a)]|| \\
&= ||\mathbf{E}[\bar{R}(s,a) + \gamma q_n(s',a') - \bar{R}(s,a) - \gamma q_{\text{AV-TD}}(s',a')]|| \\
&\leq \gamma \max_s \max_a |q_n(s,a) - q_{\text{AV-TD}}(s,a)| \\
&= \gamma ||\Delta_n(s,a)||
\end{aligned}$$

which is of the form of the third condition of Lemma 4 where $\kappa = \gamma$ and $c_t = 0$ for all t, which results in meeting the third condition of Lemma 4. Finally, the fourth condition of Lemma 4, $\text{Var}[F_t(x)|P_t] \leq K(1 + \kappa||\Delta_t||^2)$, for some constant, K, is met since F_t depends at most linearly on $q_n(s,a)$, which in turn depends on the *bounded* reward function (according to our third assumption of the theorem) (Jaakkola et al., 1993). Meeting this last condition allows us to apply Lemma 4, which shows that the batch action-value TD update rule given by Equation (2) will converge to the fixed-point given by Equation (3) □

Theorem 2 (Batch expected action-value TD Convergence)**.** *Batch expected action-value TD(0), as defined by update (4), converges to the action-value function given by (5) when the following assumptions hold:*

1. *$\mathcal{S}$ and $\mathcal{A}$ are finite.*
2. *$\sum_n \beta_n(s,a) = \infty$, $\sum_n (\beta_n(s,a))^2 < \infty$.*
3. *The reward function is bounded.*

Proof. The proof strategy is as follows: we want to apply Lemma 4 to show convergence to Equation (5). In order to apply Lemma 4, the four conditions need to be met. The first and second conditions of Lemma 4 are met by the first two assumptions of the theorem. We discuss meeting the fourth condition at the end of the proof. For the third condition of Lemma 4, we must adapt Equation (4) so that is applicable. We define $\hat{c}(s,a,s')$ as the number of occurrences of transition tuple (s,a,s'). Notice that $\hat{c}(s,a,s') = \hat{c}(s)\hat{\pi}(a|s)\hat{P}(s'|s,a)$.

Consider the update rule of batch expected action-value TD for policy evaluation on the $n+1^{\text{th}}$ iteration of processing the batch of data:

$$\begin{aligned}
q_{n+1}(s,a) &= q_n(s,a) + \alpha \sum_{s'\in\widehat{\mathcal{S}}} \hat{c}(s,a,s')[\bar{R}(s,a) + \gamma \sum_{a'\in\mathcal{A}} \pi(a'|s')q_n(s',a') - q_n(s,a)] \\
&= q_n(s,a) + \alpha\hat{c}(s,a) \sum_{s'\in\widehat{\mathcal{S}}} \hat{P}(s'|s,a)[\bar{R}(s,a) + \gamma \sum_{a'\in\mathcal{A}} \pi(a'|s')q_n(s',a') - q_n(s,a)] \\
&= q_n(s,a) + \beta_n(s,a) \sum_{s'\in\widehat{\mathcal{S}}} \hat{P}(s'|s,a)[\bar{R}(s,a) + \sum_{a'\in\mathcal{A}} \pi(a'|s')q_n(s',a') - q_n(s,a)] \qquad \beta_n(s,a) = \alpha\hat{c}(s,a)
\end{aligned}$$

where $\beta_n(s,a) = \alpha\hat{c}(s,a)$. We now subtract $q_{\text{EXP-AV-TD}}(s,a)$ (Equation (5)) from both sides.

$$\begin{aligned}
q_{n+1}(s,a) - q_{\text{EXP-AV-TD}}(s,a) = q_n(s,a) - q_{\text{EXP-AV-TD}}(s,a)& \\
+ \beta_n(s,a) \sum_{s'\in\widehat{\mathcal{S}}} \hat{P}(s'|s,a)[\bar{R}(s,a) + \sum_{a'\in\mathcal{A}} \pi(a'|s')q_n(s',a') - q_n(s,a)]&
\end{aligned}$$

Then adding and subtracting $\beta_n(s,a)\sum_{s'\in\widehat{\mathcal{S}}} \hat{P}(s'|s,a)q_{\text{EXP-AV-TD}}(s,a)$ on RHS. We also define $\Delta_n(s,a) = q_n(s,a) - q_{\text{EXP-AV-TD}}(s,a)$.

$$\begin{aligned}
\Delta_{n+1}(s,a) &= \Delta_n(s,a) + \beta_n(s,a) \sum_{s'\in\widehat{\mathcal{S}}} \hat{P}(s'|s,a)[\bar{R}(s,a) + \sum_{a'\in\mathcal{A}} \pi(a'|s')q_n(s',a') - q_n(s,a)] \\
&\quad + \beta_n(s,a) \sum_{s'\in\widehat{\mathcal{S}}} \hat{P}(s'|s,a)q_{\text{EXP-AV-TD}}(s,a) - \beta_n(s,a) \sum_{s'\in\widehat{\mathcal{S}}} \hat{P}(s'|s,a)q_{\text{EXP-AV-TD}}(s,a) \\
&= (1 - \beta_n(s,a) \sum_{s'\in\widehat{\mathcal{S}}} \hat{P}(s'|s,a))\Delta_n(s,a) \\
&\quad + \beta_n(s,a) \sum_{s'\in\widehat{\mathcal{S}}} \hat{P}(s'|s,a)[\bar{R}(s,a) + \sum_{a'\in\mathcal{A}} \pi(a'|s')q_n(s',a') - q_{\text{EXP-AV-TD}}(s,a)] \\
\Delta_{n+1}(s,a) &= (1 - \beta_n(s,a))\Delta_n(s,a) + \beta_n(s,a) \sum_{s'\in\widehat{\mathcal{S}}} \hat{P}(s'|s,a)[\bar{R}(s,a) + \sum_{a'\in\mathcal{A}} \pi(a'|s')q_n(s',a') - q_{\text{EXP-AV-TD}}(s,a)]
\end{aligned}$$

where $\sum_{s'\in\widehat{\mathcal{S}}} \hat{P}(s'|s,a) = 1$.

Then by comparing $\Delta_{n+1}(s,a)$ with Lemma 4, we have $\zeta_t = \beta_n$ and $F_t = \sum_{s'\in\widehat{\mathcal{S}}} \hat{P}(s'|s,a)[\bar{R}(s,a) + \gamma\sum_{a'\in\mathcal{A}} \pi(a'|s')q_n(s',a') - q_{\text{EXP-AV-TD}}(s,a)]$. Then for the third condition of Lemma 4,

$$\begin{aligned}
||\mathbf{E}[F_t|P_t]|| &= ||\mathbf{E}[\bar{R}(s,a) + \gamma \sum_{a'\in\mathcal{A}} \pi(a'|s')q_n(s',a') - q_{\text{EXP-AV-TD}}(s,a)]|| \\
&= ||\mathbf{E}[\bar{R}(s,a) + \gamma \sum_{a'\in\mathcal{A}} \pi(a'|s')q_n(s',a') - \bar{R}(s,a) - \gamma \sum_{a'\in\mathcal{A}} \pi(a'|s')q_{\text{EXP-AV-TD}}(s',a')]|| \\
&\leq \gamma \max_s \max_a |q_n(s,a) - q_{\text{EXP-AV-TD}}(s,a)| \\
&= \gamma||\Delta_n(s,a)||
\end{aligned}$$

which is of the form of the third condition of Lemma 4 where $\kappa = \gamma$ and $c_t = 0$ for all t, which results in meeting the third condition of Lemma 4. Finally, the fourth condition of Lemma 4, $\text{Var}[F_t(x)|P_t] \leq K(1 + \kappa||\Delta_t||^2)$, for some constant, K, is met since F_t depends at most linearly on $q_n(s,a)$, which in turn depends on the *bounded* reward function (according to our third assumption of the theorem) (Jaakkola et al., 1993). Meeting this last condition allows us to apply Lemma 4, which shows that the batch expected action-value TD update rule given by Equation (4) will converge to the fixed-point given by Equation (5) □

Theorem 3 (Batch PSEC action-value TD Convergence)**.** *Batch PSEC action-value TD, as defined by update (6), converges to the action-value function given by (7) when the following assumptions hold:*

1. *$\mathcal{S}$ and $\mathcal{A}$ are finite.*
2. *$\sum_n \beta_n(s,a) = \infty$, $\sum_n (\beta_n(s,a))^2 < \infty$.*
3. *The reward function is bounded.*

Proof. The proof strategy is as follows: we want to apply Lemma 4 to show convergence to Equation (7). In order to apply Lemma 4, the four conditions need to be met. The first and second conditions of Lemma 4 are met by the first two assumptions of the theorem. We discuss meeting the fourth condition at the end of the proof. For the third condition of Lemma 4, we must adapt Equation (6) so that is applicable. We define $\hat{c}(s,a,s',a')$ as the number of occurrences of transition tuple (s,a,s',a'). Notice that $\hat{c}(s,a,s',a') = \hat{c}(s)\hat{\pi}(a|s)\hat{P}(s'|s,a)\hat{\pi}(a'|s,a,s')$. Let $\frac{\pi(a'|s')}{\hat{\pi}(a'|s,a,s')}$ be the PSEC correction ratio. For this theorem, we make the assumption that all (s,a,s',a') transition tuples have been visited.

Consider the update rule of batch PSEC action-value TD for policy evaluation on the $n+1^{\text{th}}$ iteration of processing the batch of data:

$$\begin{aligned}
q_{n+1}(s,a) &= q_n(s,a) + \alpha \sum_{s' \in \widehat{\mathcal{S}}} \sum_{a' \in \widehat{\mathcal{A}}} \hat{c}(s,a,s',a')[\frac{\pi(a'|s')}{\hat{\pi}(a'|s,a,s')}(\bar{R}(s,a) + \gamma q_n(s',a')) - q_n(s,a)] \\
&= q_n(s,a) \\
&\quad + \alpha \hat{c}(s,a) \sum_{s' \in \widehat{\mathcal{S}}} \sum_{a' \in \widehat{\mathcal{A}}} \hat{P}(s'|s,a)\hat{\pi}(a'|s,a,s')[\frac{\pi(a'|s')}{\hat{\pi}(a'|s,a,s')}(\bar{R}(s,a) + \gamma q_n(s',a')) - q_n(s,a)] \\
&= q_n(s,a) \\
&\quad + \beta_n(s,a) \sum_{s' \in \widehat{\mathcal{S}}} \sum_{a' \in \widehat{\mathcal{A}}} \hat{P}(s'|s,a)\hat{\pi}(a'|s,a,s')[\frac{\pi(a'|s')}{\hat{\pi}(a'|s,a,s')}(\bar{R}(s,a) + \gamma q_n(s',a')) - q_n(s,a)]
\end{aligned}$$

where $\beta_n(s,a) = \alpha \hat{c}(s,a)$. We now subtract $q_{\text{PSEC-AV-TD}}(s,a)$ (Equation (7)) from both sides.

$$\begin{aligned}
q_{n+1}(s,a) - q_{\text{PSEC-AV-TD}}(s,a) &= q_n(s,a) - q_{\text{PSEC-AV-TD}}(s,a) \\
&\quad + \beta_n(s,a) \sum_{s' \in \widehat{\mathcal{S}}} \sum_{a' \in \widehat{\mathcal{A}}} \hat{P}(s'|s,a)\hat{\pi}(a'|s,a,s')[\frac{\pi(a'|s')}{\hat{\pi}(a'|s,a,s')}(\bar{R}(s,a) + \gamma q_n(s',a')) - q_n(s,a)]
\end{aligned}$$

Then adding and subtracting $\beta_n(s,a) \sum_{s' \in \widehat{\mathcal{S}}} \sum_{a' \in \widehat{\mathcal{A}}} \hat{P}(s'|s,a)\pi(a'|s')q_{\text{PSEC-AV-TD}}(s,a)$ on RHS. We also define $\Delta_n(s,a) = q_n(s,a) - q_{\text{PSEC-AV-TD}}(s,a)$.

$$
\begin{aligned}
\Delta_{n+1}(s,a) &= \Delta_n(s,a) \\
&+ \beta_n(s,a) \sum_{s'\in\widehat{\mathcal{S}}} \sum_{a'\in\widehat{\mathcal{A}}} \hat{P}(s'|s,a)\hat{\pi}(a'|s,a,s')[\frac{\pi(a'|s')}{\hat{\pi}(a'|s,a,s')}(\bar{R}(s,a) + \gamma q_n(s',a')) - q_n(s,a)] \\
&+ \beta_n(s,a) \sum_{s'\in\widehat{\mathcal{S}}} \sum_{a'\in\widehat{\mathcal{A}}} \hat{P}(s'|s,a)\pi(a'|s') q_{\text{PSEC-AV-TD}}(s,a) \\
&- \beta_n(s,a) \sum_{s'\in\widehat{\mathcal{S}}} \sum_{a'\in\widehat{\mathcal{A}}} \hat{P}(s'|s,a)\pi(a'|s') q_{\text{PSEC-AV-TD}}(s,a) \\
&= \Delta_n(s,a) - \beta_n(s,a) q_n(s,a) + \beta_n(s,a) \sum_{s'\in\widehat{\mathcal{S}}} \sum_{a'\in\widehat{\mathcal{A}}} \hat{P}(s'|s,a)\pi(a'|s')(\bar{R}(s,a) + \gamma q_n(s',a')) \\
&+ \beta_n(s,a) \sum_{s'\in\widehat{\mathcal{S}}} \sum_{a'\in\widehat{\mathcal{A}}} \hat{P}(s'|s,a)\pi(a'|s') q_{\text{PSEC-AV-TD}}(s,a) \\
&- \beta_n(s,a) \sum_{s'\in\widehat{\mathcal{S}}} \sum_{a'\in\widehat{\mathcal{A}}} \hat{P}(s'|s,a)\pi(a'|s') q_{\text{PSEC-AV-TD}}(s,a) \\
&= (1-\beta_n(s,a))\Delta_n(s,a) \\
&+ \beta_n(s,a) \sum_{s'\in\widehat{\mathcal{S}}} \sum_{a'\in\widehat{\mathcal{A}}} \hat{P}(s'|s,a)\pi(a'|s')[\bar{R}(s,a) + \gamma q_n(s',a') - q_{\text{PSEC-AV-TD}}(s,a)] \qquad \text{Note below} \\
\Delta_{n+1}(s,a) &= (1-\beta_n(s,a))\Delta_n(s,a) \\
&+ \beta_n(s,a) \sum_{s'\in\widehat{\mathcal{S}}} \sum_{a'\in\widehat{\mathcal{A}}} \hat{P}(s'|s,a)\pi(a'|s')[\bar{R}(s,a) + \gamma q_n(s',a') - q_{\text{PSEC-AV-TD}}(s,a)]
\end{aligned}
$$

Note that above, we used the fact that $\sum_{s'\in\widehat{\mathcal{S}}} \hat{P}(s'|s,a) \sum_{a'\in\widehat{\mathcal{A}}} \pi(a'|s') = 1$ when all (s,a,s',a') tuples have been visited and $\sum_{s'\in\widehat{\mathcal{S}}} \sum_{a'\in\widehat{\mathcal{A}}} \hat{P}(s'|s,a)\hat{\pi}(a'|s,a,s') = 1$.

Then by comparing $\Delta_{n+1}(s,a)$ with Lemma 4, we have $\zeta_t = \beta_n$ and $F_t = \sum_{s'\in\widehat{\mathcal{S}}} \sum_{a'\in\widehat{\mathcal{A}}} \hat{P}(s'|s,a)\pi(a'|s')[\bar{R}(s,a) + \gamma q_n(s',a') - q_{\text{PSEC-AV-TD}}(s,a)] = \bar{R}(s,a) - q_{\text{PSEC-AV-TD}}(s,a) + \gamma \sum_{s'\in\widehat{\mathcal{S}}} \sum_{a'\in\widehat{\mathcal{A}}} \hat{P}(s'|s,a)\pi(a'|s') q_n(s',a')$, where again, we assume that all (s,a,s',a') tuples have been visited. Then for the third condition of Lemma 4,

$$
\begin{aligned}
||\mathbf{E}[F_t|P_t]|| &= ||\mathbf{E}[\bar{R}(s,a) + \gamma \sum_{a'\in\widehat{\mathcal{A}}} \pi(a'|s') q(s',a') - q_{\text{PSEC-AV-TD}}(s,a)]|| \\
&= ||\mathbf{E}[\bar{R}(s,a) + \gamma \sum_{a'\in\widehat{\mathcal{A}}} \pi(a'|s') q_n(s',a') - \bar{R}(s,a) - \gamma \sum_{a'\in\widehat{\mathcal{A}}} \pi(a'|s') q_{\text{PSEC-AV-TD}}(s',a')]|| \\
&\leq \gamma \max_s \max_a |q_n(s,a) - q_{\text{PSEC-AV-TD}}(s,a)| \\
&= \gamma ||\Delta_n(s,a)||
\end{aligned}
$$

which is of the form of the third condition of Lemma 4 where $\kappa = \gamma$ and $c_t = 0$ for all t, which results in meeting the third condition of Lemma 4. Finally, the fourth condition of Lemma 4, $\text{Var}[F_t(x)|P_t] \leq K(1 + \kappa||\Delta_t||^2)$, for some constant, K, is met since F_t depends at most linearly on $q_n(s,a)$, which in turn depends on the *bounded* reward function (according to our third assumption of the theorem) (Jaakkola et al., 1993). Meeting this last condition allows us to apply Lemma 4, which shows that the batch PSEC action-value TD update rule given by Equation (6) will converge to the fixed-point given by Equation (7) □

B Batch Action-Value TD and Batch Expected Action-Value TD Pseudocode

Algorithm 1 Batch action-value TD and Batch expected action-value TD to estimate q^π

```
1: Input: policy to evaluate $\pi$, batch $\mathcal{D}$, step-size $\alpha > 0$, convergence threshold $\epsilon > 0$
2: Initialize: $q(s, a)$ arbitrarily $\forall s \in \mathcal{S}, \forall a \in \mathcal{A}$, aggregation of action-values, $u(s, a) := 0, \forall s \in \mathcal{S}, \forall a \in \mathcal{A}$, batch process counter, $i = 0$
3: **while** $|q_{i+1} - q_i| \geq \epsilon\mathbf{1}$ **do**
4:   **for** each transition, $(S, A, R, S', A') \in \mathcal{D}$ **do**
5:     **if** expected action-value TD **then**
6:       $y \leftarrow R + \gamma \sum_{a' \in \mathcal{A}} \pi(a'|S)q(S', a')$
7:     **else**
8:       $y \leftarrow R + \gamma q(S', A')$
9:     **end if**
10:    $u(S, A) \leftarrow u(S, A) + [y - q(S, A)]$
11:  **end for**
12:  $q_{i+1} \leftarrow q_i + \alpha U$ {batch update}
13:  $u(s, a) := 0, \forall s \in \mathcal{S}, \forall a \in \mathcal{A}$ {clear aggregation}
14:  $i \leftarrow i + 1$ {update batch process counter}
15: **end while**
```

C Extended Empirical Description

In this appendix we provide additional details for our empirical evaluation.

C.1 Gridworld

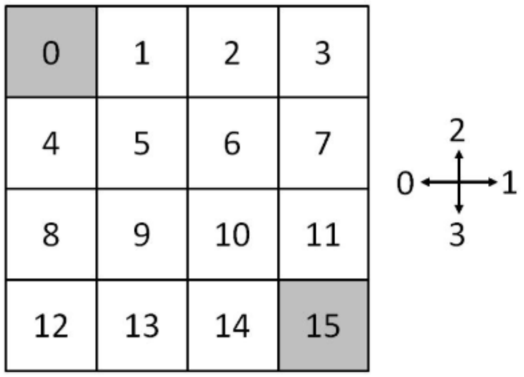

Figure 2: The Gridworld environment. Start at top left, bottom right is terminal state, discrete action space consists of the cardinal directions, and discrete state space is the location in the grid. This specific image was taken from this link.

This domain is a 4×4 grid, where an agent starts at $(0, 0)$ and tries to navigate to $(3, 3)$. The states are the discrete positions in the grid and actions are the 4 cardinal directions. The reward function is 100 for reaching $(3, 3)$, -10 for reaching $(1, 1)$, 1 for reaching $(1, 3)$, and -1 for reaching all other states. If an agent takes an action that hits a wall, the agent stays in the same location. The transition dynamics are controlled by a parameter, p, where with probability p, an agent takes the intended action, else it takes an adjacent action with probability $(1 - p)/2$. All policies use a softmax action selection distribution with value θ_{sa}, for each state, s, and action a. The probability of taking action a in state s is given by:

$$\pi(a|s) = \frac{e^{\theta_{sa}}}{\sum_{a' \in \mathcal{A}} e^{\theta_{sa'}}}$$

The fixed policy used was an equiprobable policy in each cardinal direction.

For the comparisons of batch action-value TD, batch expected action-value TD, and batch PSEC action-value TD, we conducted a parameter sweep of the learning rates for the varying batch sizes. The parameter sweep was over: $\{5e^{-3}, 1e^{-3}, 5e^{-2}, 1e^{-2}, 5e^{-1}\}$. We used a action-value convergence threshold of $1e^{-5}$.