Inverse Policy Evaluation for Value-based Decision Making

Alan Chan[*]
Department of Computer Science
Université de Montréal
Montréal, QC
alan.chan@mila.quebec

Kristopher De Asis*
Department of Computing Science
University of Alberta
Edmonton, AB
kldeasis@ualberta.ca

Richard S. Sutton
Department of Computing Science
University of Alberta
Edmonton, AB
rsutton@ualberta.ca

Abstract

Value-based methods for control often involve approximate value iteration (e.g., Q-learning), and behaving greedily with respect to the resulting value estimates with some degree of entropy to ensure the state-space is sufficiently explored. Such a greedy policy is an improvement over the policy of which the current values reflect. As learning progresses, value-iteration may produce value functions that do not correspond with *any* policy. This is especially relevant with function-approximation, when the true value function cannot be perfectly represented. This raises questions about what such inaccurate value functions represent, and whether there exists alternative ways to derive value-based behavior. In this work, we explore the use of *Inverse Policy Evaluation*, the process of solving for a likely policy given a value function. We derive a simple, incremental algorithm for the procedure, analyze how inaccuracies in the value function manifest in the corresponding policy, and provide empirical results emphasizing key properties of the mapping from value functions to policies. This sets up a framework for matching *proposed returns*, as opposed to explicit maximization. Coupling this procedure with value-iteration, this provides a novel approach for deriving behavior from a value function which also tends toward an optimal policy, but appears to account for estimation error in the resulting behavior.

Keywords: reinforcement Learning, value-based control, exploration

Acknowledgements

The authors thank the entire Reinforcement Learning and Artificial Intelligence research group for providing the environment to nurture and support this research. We gratefully acknowledge funding from Alberta Innovates – Technology Futures, the Alberta Machine Intelligence Institute, Google Deepmind, and from the Natural Sciences and Engineering Research Council of Canada.

[*] = Equal contribution.

1 Value-based Reinforcement Learning

Reinforcement learning (RL) formalizes the sequential decision-making problem with the Markov Decision Process (MDP) framework [14, 17]. At each discrete time step t, an agent observes the current state $S_t \in \mathcal{S}$, where $\mathcal{S}$ is the set of states in an MDP. Given S_t, an agent selects action $A_t \in \mathcal{A}(S_t)$, where $\mathcal{A}(s)$ is the set of available actions in state s. The environment then returns a reward $R_{t+1} \in \mathbb{R}$, and an observation of the next state $S_{t+1} \in$, sampled from the environment's transition dynamics: $p(s', r|s, a) = \Pr(S_{t+1} = s', R_{t+1} = r|S_t = s, A_t = a)$. An agent selects actions according to a policy $\pi(a|s) = \Pr(A_t = a|S_t = s)$, and its goal is to find an *optimal policy* π^* which from each state, maximizes the expected discounted sum of future rewards with discount factor γ (denoted the expected *return*).

Value-based methods estimate *value functions* which quantifies a policy's performance. In control, an *action-value function* is approximated, representing the expected return from starting in state s, taking action a, and following policy π thereafter:

$$q_\pi(s, a) = \mathbb{E}_\pi\Big[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}\Big|S_t = s, A_t = a\Big] \tag{1}$$

Computing a policy's value function is known as *policy evaluation*. Value-based methods then rely on *policy improvement*, where greedifying with respect to another policy's value function will produce an improved policy. *Policy iteration* [4] interleaves policy evaluation and improvement until an optimal policy is found. In contrast, *policy gradient* methods explicitly parameterize a policy, and updates the parameters to maximize an objective [18, 17].

Q-learning [22] is a popular value-based method which ties to directly estimate the value function of the optimal policy. Value functions can be expressed in terms of successor states' values through their *Bellman equations*. For q_π, we have:

$$q_\pi(s, a) = \sum_{s', r} p(s', r|s, a)\Big(r + \gamma \sum_{a'} \pi(a'|s') q_\pi(s', a')\Big) \tag{2}$$

When π is greedy with respect to value estimates, the latter term becomes the maximum action-value in the successor state, and gives the *Bellman optimality equation*, the recursive relationship for the optimal action-value function q_*. Repeat evaluation of the Bellman optimality equation across the state-space is known as *value-iteration*, and is akin to policy iteration with partial policy evaluation. Q-learning uses stochastic samples of p to perform approximate value-iteration:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\Big(R_{t+1} + \gamma \max_{a'} Q(S_{t+1}, a') - Q(S_t, A_t)\Big) \tag{3}$$

with step size α. Convergence of Q to q_* for every state-action pair requires that all state-action pairs are visited infinitely often [5]. Even if convergence is not guaranteed, as is the case with function approximation [3, 19, 1], it is still necessary to *explore* different parts of state-space to obtain a reasonable estimate of q_* to inform decision making.

This exploration requirement is often satisfied with ϵ-greedy action selection [11, 17], where an agent behaves greedily with respect to its current estimates with probability $1 - \epsilon$, and behaves uniform randomly otherwise. However, there are practical concerns with ϵ-greedy behavior: 1) reliance on random exploration is likely inefficient in large state spaces [9]; 2) due to estimation error or representational capacities, the greedy action may be a poor choice [7]; 3) needing to specify ϵ as a fixed value or an annealing schedule; and 4) the non-smoothness of the behavior policy with respect to changes in the value function can result in non-convergence [12, 13, 21].

Several alternatives to ϵ-greedy have been explored, e.g., Boltzmann policies [2], conservative policy iteration [8, 20]. Many of them directly work with the policy that an agent aims to *eventually* evaluate, i.e., some modification of the greedy policy. No work to our knowledge has explicitly considered the policy which corresponds with the *current* value estimates. Such an approach would preferably take estimation error into account, e.g., due to insufficient exploration or function approximation errors, rather than assume the current values are accurate. Given $Q = q_*$, the policy that gives zero Bellman error *is* an optimal policy by the uniqueness of the solution of the Bellman optimality equation [5]. Should Q be inaccurate, the policy should be suboptimal in a way that's directly related to the errors.

2 Inverse Policy Evaluation

Inverse Policy Evaluation (IPE) aims to derive a behavior policy that is consistent with a value function in the following sense:

$$\pi \in \operatorname*{argmin}_{\pi \in \Pi} \|Q(s, a) - \mathbb{E}_{s', a'}[r(s, a) + \gamma Q(s', a')]\| \tag{4}$$

where $s' \sim p(\cdot|s, a)$, $a' \sim \pi(a'|s')$, $r(s, a)$ giving the expected immediate reward for s, a, and $\|\cdot\|$ being some fixed norm over state-action pairs. Let us solidify the intuition that the solution takes function approximation error into account.

Proposition 1. *Assume that we are trying to estimate q_π with Q, for some q_π. Denote the solution of Equation 4 by π_{IPE}, the Bellman operator of Equation 2 as $\mathcal{T}$, and let $\|\cdot\|$ denote any norm under which Bellman operators are contraction mappings (e.g., infinity norm). We have the following bound.*

$$\|q_\pi - q_{\pi_{IPE}}\| \leq \frac{1}{1-\gamma}\big((1+\gamma)\|q_\pi - Q\| + \|\mathcal{T}^{\pi_{IPE}}Q - Q\|\big)$$

The first norm on the right-hand side measures the estimation error of Q, and the second norm on the right-hand side is exactly the objective in Equation 4. From this, the return generated by π_{IPE} is close to the return generated by π, proportional to how close Q is to q_π. We now consider how changes in value estimates affect π_{IPE}.

Proposition 2. *Suppose Q_1, Q_2 are two approximate action-value functions. Let π_i denote the IPE solution of Q_i. Then*

$$\|q_{\pi_1} - q_{\pi_2}\| \leq \frac{1}{1-\gamma}((1+\gamma)\|Q_1 - Q_2\| + \|\mathcal{T}^{\pi_1}Q_1 - Q_1\| + \|\mathcal{T}^{\pi_2}Q_2 - Q_2\|)$$

This smoothness result for IPE contrasts ϵ-greedy policies, which are known to be non-smooth with respect to changes in action-value estimates [12, 13]. So IPE can produce an estimation-error-aware policy, but how do we compute this? With the ℓ_2 norm and policy π_θ smoothly parameterized by θ, we can approach π_{IPE} through the gradient-based update:

$$\delta = R_{t+1} + \gamma \sum_{a'} \pi_\theta(a'|S_{t+1})Q(S_{t+1}, a') - Q(S_t, A_t)$$
$$\theta_{t+1} \leftarrow \theta_t - \alpha 2\delta\gamma \sum_{a'} \nabla_\theta \pi_\theta(a'|S_{t+1})Q(S_{t+1}, a') \tag{5}$$

where δ is the conventional *temporal difference (TD) error* [16]. Of note, the update is remarkably similar to the *all-actions policy gradient* update [18, 17]. One can interpret Equation 5 as a policy gradient update for *matching proposed returns* of an approximate value function Q, with δ changing signs to ensure that the return is matched, rather than maximized.

It may be the case that Q does not correspond with *any* policy, relating to *delusional bias* [6, 10]. Nevertheless, the resulting policy will still minimize Equation 4, which by Proposition 1 would be close in a sense of achieving similar returns as a nearby "valid" value function (i.e., has a corresponding policy).

Should we couple IPE with value-iteration, the proposed returns will tend toward those achieved by the optimal policy, and IPE will approach an optimal policy. This results in an actor-critic-like procedure, but with less cyclic dependencies. Actor-critic methods typically perform a cycle of: evaluating the current behavior policy, improving the policy with the current evaluation. In contrast, value-iteration tries to compute the largest possible returns (and not those of the current policy), that the remaining dependency is on the distribution of transitions experienced by the current policy.

3 Empirical Evaluation

Here we run some experiments to validate some key intuitions behind IPE.

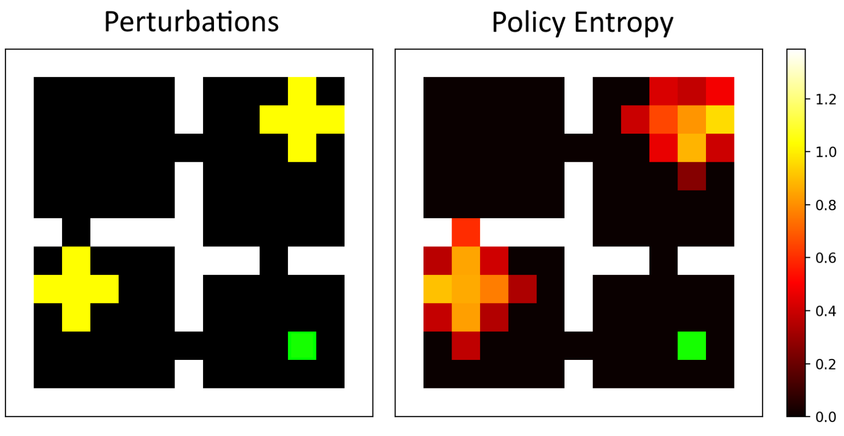

Figure 1: Resulting π_{IPE} entropy after perturbing q_* in the Four Rooms domain. Green denotes the goal state.

3.1 Stochasticity from Estimation Error

Using the Four Rooms domain [15], we look at how estimation error impacts the stochasticity of π_{IPE}. First, we compute q_* for the domain, and initialize π_{IPE} to an optimal policy (breaking ties consistently). We then perturb q_* in select state-action pairs by samples from $\sim \mathcal{N}(0, 0.1)$, and run IPE to convergence. Averaged over 1000 runs, Figure 1 visualizes the policy's entropy in each state of the environment and how they relate to the perturbations in the value function.

It can be seen that the increase in policy entropy is directly related to the perturbed state-action pairs, affecting both the perturbed areas as well as states immediately transitioning to them. This validates that the policy becomes suboptimal in a way that's related to the errors. It further motivates the possible use of IPE to adapt exploration, as the stochasticity tends to explore areas with larger errors. Viewing the policy entropy as a level of confidence in the accuracy of value estimates, it may also inform parameter selection, as having the confidence expressed in entropy is convenient for hyper-parameters with probabilistic interpretations (e.g., λ in TD(λ), ϵ in ϵ-greedy). Along these lines, we consider an ϵ-greedy Q-learning agent which additionally performs IPE updates on the experienced transitions, and adapts ϵ to match the entropy of the estimated π_{IPE} in the current state. We denote this approach ϵ-IPE.

3.2 Balancing Reward and Estimation Error

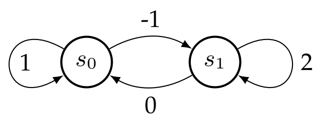

Figure 2: The switch-stay MDP. All transitions are deterministic and the agent starts in state s_0.

Here we use a simple 2-state MDP detailed in Figure 2. We compare four Q-learning agents, each differing in how behavior is derived: (1) ϵ-greedy with fixed ϵ, (2) ϵ-greedy with annealing ϵ, (3) following π_{IPE}, and (4) ϵ-IPE. We swept over ϵ for (1), the number of steps to linearly anneal ϵ from 1.0 to 0.1 for (2), and the IPE step size, α_π, for (3) and (4). Each setting performed 1000 runs of 500 steps, and Figure 3 shows the average reward over the 500 steps, as well as the final root-mean-squared error (RMSE) in the approximated optimal value function.

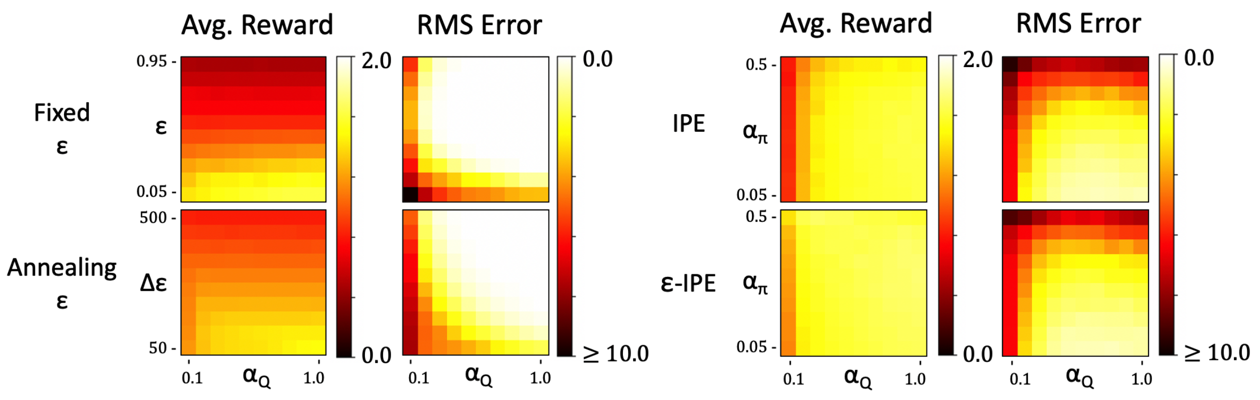

Figure 3: Hyperparameter sensitivities on the switch-stay MDP. 1000 runs of 500 steps were performed for each behavior policy's hyperparameter configuration.

In Figure 3, Q-learning with ϵ-greedy exhibited a negative correlation between the average reward and the value function RMSE. To attain high average reward, the agent tends to settle for an *inaccurate* value function, and vice-versa. Such a relation is what one might intuitively expect from tending toward either extreme of the exploration-exploitation trade-off.

On the other hand, IPE and ϵ-IPE exhibit a positive correlation between value function accuracy and the average reward obtained (with respect to the behavior policy's parameter). Given the large overlap between the regions of high average reward and low RMSE, there seems to be, without careful parameter tuning, a natural adequate balance of (1) exploration needed to learn an accurate value function and (2) exploitation of value estimates to achieve a large expected return.

4 Discussion and Conclusions

Our results suggest that IPE, when combined with value-iteration, provides a novel, viable way to derive a sensible behavior policy for value-based control. While greedy policies generally lead to (immediate) larger returns [8], we highlight how such greedification is still present in the use of value-iteration. We further emphasize how stochasticity related to estimation error can lead to visitation distributions which explore more in regions with larger estimation errors (and vice-versa), and be a useful metric of confidence in value estimates to inform parameter selection.

A key limitation in this work is the focus on the tabular setting. We expect the intuitions and results surrounding estimation errors to still be relevant, as examining behavior under inaccurate value functions (or perturbations of the optimal values) can be seen as simulating the inability to perfectly represent the value function. That said, extensive evaluation with function approximation (both linear and non-linear) are warranted to assess the approach's scalability.

References

[1] J. Achiam, E. Knight, and P. Abbeel. Towards characterizing divergence in deep q-learning. *arXiv preprint arXiv:1903.08894*, 2019.

[2] K. Asadi and M. L. Littman. An alternative softmax operator for reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 243–252. JMLR. org, 2017.

[3] L. Baird. Residual algorithms: Reinforcement learning with function approximation. In *Machine Learning Proceedings 1995*, pages 30–37. Elsevier, 1995.

[4] D. P. Bertsekas. Approximate policy iteration: A survey and some new methods. *Journal of Control Theory and Applications*, 9(3):310–335, 2011.

[5] D. P. Bertsekas and J. N. Tsitsiklis. *Neuro-dynamic programming*, volume 5. Athena Scientific Belmont, MA, 1996.

[6] R. Dadashi, A. A. Taïga, N. L. Roux, D. Schuurmans, and M. G. Bellemare. The value function polytope in reinforcement learning. *arXiv preprint arXiv:1901.11524*, 2019.

[7] H. V. Hasselt. Double q-learning. In *Advances in neural information processing systems*, pages 2613–2621, 2010.

[8] S. Kakade and J. Langford. Approximately optimal approximate reinforcement learning. In *ICML*, volume 2, pages 267–274, 2002.

[9] D. Korenkevych, A. R. Mahmood, G. Vasan, and J. Bergstra. Autoregressive policies for continuous control deep reinforcement learning. *arXiv preprint arXiv:1903.11524*, 2019.

[10] T. Lu, D. Schuurmans, and C. Boutilier. Non-delusional q-learning and value-iteration. In *Advances in neural information processing systems*, pages 9949–9959, 2018.

[11] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[12] T. J. Perkins and M. D. Pendrith. On the existence of fixed points for q-learning and sarsa in partially observable domains. In *ICML*, pages 490–497, 2002.

[13] T. J. Perkins and D. Precup. A convergent form of approximate policy iteration. In *Advances in neural information processing systems*, pages 1627–1634, 2003.

[14] M. L. Puterman. *Markov Decision Processes.: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 2014.

[15] R. Sutton, D. Precup, and S. Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112:181–211, 1999.

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

[17] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[18] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063, 2000.

[19] J. N. Tsitsiklis and B. Van Roy. Analysis of temporal-diffference learning with function approximation. In *Advances in neural information processing systems*, pages 1075–1081, 1997.

[20] N. Vieillard, O. Pietquin, and M. Geist. Deep conservative policy iteration. *arXiv preprint arXiv:1906.09784*, 2019.

[21] P. Wagner. Optimistic policy iteration and natural actor-critic: A unifying view and a non-optimality result. In *Advances in Neural Information Processing Systems*, pages 1592–1600, 2013.

[22] C. J. Watkins and P. Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.