Batch Reinforcement Learning with State Importance

Lihong Li, Vadim Bulitko, and Russell Greiner

University of Alberta,
Department of Computing Science,
Edmonton, Alberta, Canada T6G 2E8
{lihong,bulitko,greiner}@cs.ualberta.ca

Abstract. We investigate the problem of using function approximation in reinforcement learning where the agent's policy is represented as a classifier mapping states to actions. High classification accuracy is usually deemed to correlate with high policy quality. But this is not necessarily the case as increasing classification accuracy can actually decrease the policy's quality. This phenomenon takes place when the learning process begins to focus on classifying less "important" states. In this paper, we introduce a measure of state's decision-making importance that can be used to improve policy learning. As a result, the focused learning process is shown to converge faster to better policies[1].

1 Problem Formulation and Related Work

Reinforcement learning (RL) [11] provide a general framework for many sequential decision-making problems and has succeeded in a number of important applications. Let $\mathcal{S}$ be the state space, $\mathcal{A}$ the action set, and D the start-state distribution. A policy is a mapping from states to actions: $\pi : \mathcal{S} \mapsto \mathcal{A}$. The state- and action-value functions are denoted by $V^\pi(s)$ and $Q^\pi(s, a)$, respectively [11]. The quality of a policy π is measured by *policy value* [10]: $\mathcal{V}(\pi) = \mathbf{E}_{s_0 \sim D}\, V^\pi(s_0)$. A RL agent attempts to *learn* the optimal policy with maximal value: $\pi^* = \arg\max_\pi \mathcal{V}(\pi)$. The corresponding optimal state- and action-value functions are denoted by $V^*(s)$ and $Q^*(s, a)$, respectively.

In this paper, we focus on classification-based RL methods where a policy π is represented as a classifier labeling state s with action $\pi(s)$. Then learning a policy π is reduced to learning a classifier [4, 6, 7, 12]. Recent implementations of this idea have demonstrated promising performance in several domains by learning high-quality policies through high-accuracy classification. It should be noted, however, that in sequential decision-making the classification error is *not* the target performance measure of a reward-collecting agent. Consequently, increasing classification accuracy can actually *lower* the policy value [9]. An intuitive explanation is that not all states are equally important in terms of preferring one action to another. Therefore, the classification-based RL methods can be improved by *focusing the learning process on more important states*. The expected benefits include faster convergence to better policies.

We examine the so-called *batch reinforcement learning* in which the policy learning occurs *offline*. Such a framework is important wherever online learning is not feasible

[1] Due to space limitation, only deterministic policy with binary actions are discussed. Details and extensions can be found in [9].

J.-F. Boulicaut et al. (Eds.): ECML 2004, LNAI 3201, pp. 566–568, 2004.

(e.g., when the reward data are limited), and therefore a fixed set of experiences has to be acquired and used for offline policy learning [2, 8]. In particular, we are interested in a special case where the state space is *sparsely sampled* and the optimal action values for these sampled states are computed or at least estimated. The sampled states together with their optimal action values form the training data for batch learning: $T_{Q^*} = \{\langle s, a, Q^*(s,a)\rangle | s \in \mathcal{T} \subset \mathcal{S}, a \in \mathcal{A}\}$, where $\mathcal{T}$ is the sparsely sampled state space. The assumption of knowing the optimal action values may at first seem unrealistic. However, a technique called *full-trajectory-tree expansion* [5, 8] can be used to compute or estimate such values. This technique is especially useful in domains where good policies generalize well across problems of different sizes: the agent can first obtain a good policy on problems with tractable state space where the technique is applicable, and then generalize the policy to larger problems.

With the training data T_{Q^*}, the optimal actions can be computed: $\forall s \in \mathcal{T}, a^*(s) = \arg\max_a Q^*(s,a)$ and the training data for learning a classifier-based policy are formed: $T_{\mathsf{CI}} = \{\langle s, a^*(s)\rangle | s \in \mathcal{T}\}$. Finally, the optimal policy is approximated by minimizing the classification error: $\widehat{\pi}^*_{\mathsf{CI}} = \arg\min_{\widehat{\pi}^*} \sum_{s\in\mathcal{S}} \mathcal{I}(\widehat{\pi}^*(s) \neq \pi^*(s))$, where $\mathcal{I}(A) = 1$ if A is true and 0 otherwise. The subscript CI (*cost-insensitive*) is in contrast to its *cost-sensitive* counterpart that will be introduced in the next section.

2 Batch Reinforcement Learning with State Importance

In contrast to the cost-insensitive algorithm outlined in the previous section, a novel RL algorithm based on *cost-sensitive* classification is proposed, which uses the state importance values as misclassification costs. As a result, the learning process focuses on important states thereby improving the convergence speed as well as the policy value.

Intuitively, a state is important from the decision-making point of view if making a wrong decision in it can have significant repercussions. Therefore, the *importance* of a state s, $G^*(s)$, is defined as: $G^*(s) = Q^*(s, a^*(s)) - Q^*(s, \bar{a}(s))$, where $a^*(s)$ is the optimal action and $\bar{a}(s)$ is the other (sub-optimal) action [2]. Similarly, the *importance of state s under policy* π, $G^*(s,\pi)$, is defined as: $G^*(s,\pi) = Q^*(s, a^*(s)) - Q^*(s, \pi(s))$. Clearly, if $\pi(s) = a^*(s)$, then $G^*(s,\pi) = 0$; otherwise, $G^*(s,\pi) = G^*(s)$.

It is desirable for the agent to approximate π^* by agreeing with it at important states. One way is to use the state importance values as the misclassification costs: $\widehat{\pi}^*_{\mathsf{CS}} = \arg\min_{\widehat{\pi}^*} \sum_{s\in\mathcal{S}} \big(G^*(s) \cdot \mathcal{I}(\widehat{\pi}^*(s) \neq \pi^*(s))\big)$. Then learning the policy is reduced to cost-sensitive classification where s is the attribute, $a^*(s)$ is the desired class label, and $G^*(s)$ is the misclassification cost. Thus, given the training data T_{Q^*}, the agent can first compute $G^*(s)$ for all states $s \in \mathcal{T}$ to form a training set: $T_{\mathsf{CS}} = \{\langle s, a^*(s), G^*(s)\rangle \mid s \in \mathcal{T}\}$, and then compute $\widehat{\pi}^*_{\mathsf{CS}}$ using cost-sensitive classification techniques.

A question of both theoretical and practical interest is whether it is preferable to solve $\widehat{\pi}^*_{\mathsf{CS}}$ as opposed to $\widehat{\pi}^*_{\mathsf{CI}}$. It is shown [9] that: (i) the policy value is lower-bounded in terms of the cost-sensitive classification error of $\widehat{\pi}^*_{\mathsf{CS}}$; however, (ii) if the cost-insensitive classification error of $\widehat{\pi}^*_{\mathsf{CI}}$ is not zero, then no matter how small the error is, the resulting policy can be arbitrarily close to the worst policy in terms of policy value. Empirical support was gained from experiments on a series of 2D grid-world domains.

[2] NB: Such a definition of $G^*(s)$ is similar to the *advantage* introduced by Baird [1].

3 Summary and Future Work

Classification-based policy acquisition is an interesting development in RL that attempts to gain a better policy by increasing the classification accuracy. However, the correlation between policy value and classification accuracy is non-monotonic as the states are not equally important. We then proposed a measure of state's decision-making importance and outlined a way to utilize such values in a class of RL problems. Advantages of such a method are supported both theoretically and empirically. The promising initial results open several avenues for future research. First, when computing resources are limited, it is possible to focus learning only on the more important states by ignoring the others. However, the extent to which such an *a priori* pruning may lead to overfitting needs to be explored. Another area for future research is an investigation of the extent to which this approach depends on the cost-sensitive classifier. In particular, it would be interesting to investigate the benefits of applying modern cost-sensitive classification techniques (e.g., cost-proportionate example weighting [13] and boosting [3]) in focused learning.

Acknowledgments. We thank Rich Sutton, Dale Schuurmans, Ilya Levner and Greg Lee for helpful discussions and other forms of help. The research is supported by the Alberta Ingenuity Center for Machine Learning, the University of Alberta, and NSERC.

References

1. Leeman Baird. Advantage updating. Technical report, Wright-Patterson Air Force Base, 1993.
2. Thomas G. Dietterich and Xin Wang. Batch value function approximation via support vectors. In *Advances in Neural Information Processing Systems 14*, volume 14, 2002.
3. Wei Fan, Salvatore J. Stolfo, Junxin Zhang, and Philip K. Chan. AdaCost: Misclassification cost-sensitive boosting. In *Proc. of the 16th Int'l Conf. on Machine Learning*, 1999.
4. Alan Fern, SungWook Yoon, and Robert Givan. Approximate policy iteration with a policy language bias. In *Advances in Neural Information Processing Systems 16*, 2004.
5. Michael Kearns, Yishay Mansour, and Andrew Ng. Approximate planning in large POMDPs via reusable trajectories. In *Advances in Neural Information Processing Systems 12*, 2000.
6. Michail Lagoudakis and Ronald Parr. Reinforcement learning as classification: Leveraging modern classifiers. In *Proc. of the* 12^{th} *Int'l Conf. on Machine Learning*, 2003.
7. John Langford and Bianca Zadrozny. Reducing T-step reinforcement learning to classification. In *Proc. of the Machine Learning Reductions Workshop*, Chicago, IL, 2003.
8. Ilya Levner and Vadim Bulitko. Machine learning for adaptive image interpretation. In *Proc. of the* 12^{th} *Innovative Applications of Artificial Intelligence Conf.*, 2004.
9. Lihong Li. Focus of attention in reinforcement learning. Master's thesis, Department of Computing Science, University of Alberta, Edmonton, Alberta, Canada, June 2004.
10. Andrew Y. Ng and Michael Jordan. PEGASUS: A policy search method for large MDPs and POMDPs. In *Proc. of the* 16^{th} *Conf. on Uncertainty in AI*, 2000.
11. Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction.* MIT Press, Cambridge, MA, March 1998.
12. SungWook Yoon, Alan Fern, and Robert Givan. Inductive policy selection for first-order MDPs. In *Proc. of the* 18^{th} *Conference on Uncertainty in AI*, 2002.
13. Bianca Zadrozny and John Langford. Cost-sensitive learning by cost-proportionate example weighting. In *Proc. of the IEEE Int'l Conf. on Data Mining*, 2003.