Gradient-Based Estimates of Return Distributions

Christos Dimitrakakis and Samy Bengio

IDIAP Research Institute, 4 Rue de Simplon, Martigny CH 1920, Switzerland
dimitrak@idiap.ch bengio@idiap.ch

Abstract. We present a general method for maintaining estimates of the distribution of parameters in arbitrary models. This is then applied to the estimation of probability distributions over actions in value-based reinforcement learning. While this approach is similar to other techniques that maintain a confidence measure for action-values, it nevertheless offers an insight into current techniques and hints at potential avenues of further research.

1 Introduction

A large number of problems in both supervised and reinforcement learning are solved with parametric methods. In this framework we attempt to approximate a function $f^*(\cdot)$ via a parameterised function $f(\theta, \cdot)$, given samples of f^*, with parameters $\theta \in \mathbb{R}^n$. We focus on incremental optimisation methods for which an optimisation operator $\mathcal{M}(C, \theta)$, where C is an appropriately defined cost, can be defined as a stochastic process that is continuous with respect to θ. We define the sequence $\{\theta\}$ as $\theta_{t+1} = \mathcal{M}(C, \theta_t)$.

In reinforcement learning, samples of f^* are generated actively. Asymptotic convergence results exist for such methods under apropriate sampling ssumptions. If we maintain a distribution of θ_t (rather than a simple vector of parameters), we may be able to use it to generate samples in an optimal sense. In this paper we explore simple gradient-based methods for measuring the accuracy of our estimates. Two cases are considered: variance estimates and gradient estimates. A naive variance estimate, arising from simple assumptions, is given and its relation to the gradient is detailed. The relation of the gradient to convergence is outlined and finally a simple gradient estimate is given.

1.1 Variance Estimates

In the general setting, for each θ_t we sample a single value M_t from $\mathcal{M}(C, \theta_t)$, where $\mathcal{M}$ is considered as a random process. In our setting we will attempt to also maintain a confidence measure for our parameters. We will attempt to do this by measuring the variance of the process at the current point θ_t.

Firstly, we assume that M_t it is bounded[1] and we attempt to estimate $\hat{E}[M_t] \approx E[M_t]$. The simplest possible estimate can be achieved by assuming that $\mathcal{M}$ is a zero-mean process, leading to $\hat{E}[M_t] = \theta_t$.

We may further assume that $\mathcal{M}$ is Lipschitz continuous with respect to θ, (a function f satisfies a Lipschitz continuity assumption in some set S if there exists $L \in \mathbb{R}$ such that $\|\nabla f(a) - \nabla f(b)\| \leq L\|a - b\|$ for all $a, b \ \in S$). Using an expone, we obtain a variance estimate of the form

$$V_{t+1} = (1 - \zeta)V_t + \zeta(\hat{E}[M_t] - \theta_{t+1})(\hat{E}[M_t] - \theta_{t+1})', \tag{1}$$

with $\zeta \in [0, 1]$. Now we may use V_t for our estimate of the variance of $\mathcal{M}(C, \theta_t)$.

Definition 11 (Naive variance estimate) *By assuming that $\mathcal{M}$ is a zero-mean process, i.e. that $E[M_t] = \theta_t$, we have:*

$$V_{t+1} = (1 - \zeta)V_t + \zeta(\theta_t - \theta_{t+1})(\theta_t - \theta_{t+1})'. \tag{2}$$

Definition 12 (Counting variance estimate) *By assuming $E[M_t] = \theta_{t+1}$, i.e. that $\mathcal{M}$ is a deterministic process, we have:*

$$V_{t+1} = (1 - \zeta)V_t. \tag{3}$$

The latter method is equivalent to a class of counting schemes whereby we increase our certainty about the mean of some random variable with each observation. With an appropriate choice for ζ such schemes can be adequate for some problems.

We may further add a small positive constant to the above updates such that the variance does not eventually reach zero, if it is desirable. In the case where we maintain a set of parameters which are updated separately (such as in tabular reinforcement learning methods, which are further discussed in Section 2.1), then it is also appropriate to maintain separate variance estimates. In the following section we discuss how such estimates are related to the convergence of the stochastic operator $\mathcal{M}$ for the case when it expresses a stochastic gradient descent step.

Relation of Variance Estimates to Convergence In the general case, estimating $|\theta - \theta^*|$, the distance to a solution, can be as difficult as determining θ^* itself. While in the general case it is not possible to determine convergence, in certain special cases it presents a manageable task. To give a simple example, when the cost surface is quadratic (i.e $C = a(\theta^* - \theta)^2$) we have $|\theta^* - \theta| = a|\nabla_\theta C|$ and the magnitude of the steps we are taking is directly related to the convergence. It is easy to show that the mean update we have defined is an approximate measure of the gradient under some conditions.

[1] For stochastic gradient methods, under the condition that the partial derivative of the cost with respect to the parameters is bounded, all M_t are bounded.

From (1), we have

$$V_{t+1} = \sum_{k=1}^{t}(1-\eta_k)V_t + \eta_k\alpha(\delta_k + e_k)'(\delta_k + e_k)$$

$$= (1-\eta_k)^t V_1 + \sum_{k=1}^{t}(1-\eta_k)^{t-k}\eta_k\alpha(\delta_k + e_k)'(\delta_k + e_k)$$

$$= (1-\eta_k)^t V_1 + \eta_k\alpha(\sum_{k=1}^{t}(1-\eta_k)^{t-k}\delta_k'\delta_k + \sum_{k=1}^{t}(1-\eta_k)^{t-k}e_k'e_k) + 2\sum_{k=1}^{t}(1-\eta_k)^{t-k}\delta_k'e_k)$$

where e_k is a noise process such as the stochastic gradient error term. For the case when $\eta_k = 1/k$ we have, with better approximation as $t \to \infty$, and if $\delta_k = C(\theta)$ for all k (i.e. when $\alpha \to 0$)

$$\text{trace}(V) \propto \|\nabla C(\theta)\|^2 + E^2[e],$$

where e is the noise term from a stochastic gradient method.

1.2 Gradient Estimates

The relation of those estimates to the gradient is of interest because of the relationship of the gradient to the distance from the minimum under certain conditions. In particular, when $\nabla^2 C(\theta)$ is positive definite, the following holds :

Lemma 11 *Let θ^* be a local minimum of C and $\theta \in S$, with $S = \{\theta : \|\theta - \theta^*\| < \delta\}$, $\delta > 0$. If there exists $m > 0$ such that*

$$m\|z\|^2 \leq z'\nabla^2 C(\theta)z, \quad \forall\ z \in \mathbb{R}^n, \tag{4}$$

then every $\theta \in S$ satisfying $\|\nabla C(\theta)\| \leq \epsilon$ also satisfies

$$\|\theta - \theta^*\| \leq \epsilon/m, \quad C(\theta) - C(\theta^*) \leq \epsilon^2/m.$$

The proof is quite straightforward and is omitted due to lack of space. We may now define a simple estimate for the gradient itself.

Definition 13 (Gradient estimate) *By using similar assumptions as in the variance estimates, we may obtain an estimate of the gradient at time t:*

$$U_{t+1} = (1-\zeta)U_t + \zeta(\hat{E}[M_t] - \theta_{t+1}) \tag{5}$$

It is perhaps of interest to note that both the naive variance estimate and the velocity variance estimates can be used to determine convergence of parameters. It is interesting to note that for gradient methods with errors, the variance estimate includes the noise term. For reinforcement learning problems with noisy rewards this is significant, because it is related to the variance of the return. If we attempt to use such convergence criteria to select actions, either estimate may prove advantageous depending on the task.

2 Action Selection

Most, if not all, reinforcement learning (RL) methods can be viewed as a combination of estimation and sampling. Given a state space $\mathcal{S}$ and an action space $\mathcal{A}$, an agent selects actions $a \in \mathcal{A}$ according to a policy $\pi : \mathcal{S} \to \mathcal{A}$. The aim of reinforcement learning is described as finding a policy π^* that maximises a utility function, for which the only available information is reward samples r_t. This is usually formulated as finding a policy $\pi^* = \{p(a|s)|(s, a) \in \mathcal{S} \times \mathcal{A}\}$ such that

$$\pi^* = \arg\max_{\pi} E[R_t|\pi], \tag{6}$$

with $R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$, where $\gamma \in [0, 1)$ is a discount parameter such that rewards far into the future are less important than closer ones.

An important subset of reinforcement learning methods is formed by value-based methods (which are the focus of [6]). These generate an evaluation for every possible action and state pair and the policy is defined in terms of this. State-action evaluations are usually noted in short-hand as $Q(s, a) = \hat{E}[R_t|s_t = s, a_t = a, \pi]$, i.e. the expected cost/return if we take action a at state s while following policy π. Value function updates typically employ temporal-difference methods, whereby parameters are adjusted in the direction of the temporal-difference error, which has the form $\delta = r_t + \gamma\hat{E}[R_{t+1}|s_{t+1}, a_t, \pi] - Q(s, a)$. In some cases parameters are adjusted according to an importance weight, which usually takes the form of an *eligibility trace* e_i, defined for each parameter θ_i.

2.1 Application of Variance Estimates to Action Values

These variance estimates can be applied with relative ease to action value reinforcement learning using either a tabular or an approximation architecture. The naive variance estimate (2) is particularly interesting because, for the tabular case, its use results in algorithm that is similar to [5]. For this reason we shall concentrate on this particular estimate, but we will also be contrasting it to a gradient-related estimate.

In the following short sections we consider the application of such an estimate to reinforcement learning; firstly in the tabular and secondly in the function approximation case. Lastly, we describe action selection mechanisms, using the developed variance estimates, that can be applied to either case.

Tabular Action Value Methods The tabular reinforcement learning case can be obtained by defining a θ for each state-action pair Q, so that we maintain separate variance estimates for each one. Then we consider that at each time step the operator sample M_t can be defined as $M_t \equiv Q_{t+1}(s, a) = Q_t(s, a) + \alpha(r_t + \hat{E}[R_{t+1}] - Q_t(s, a))$. By substituting this into (2), we obtain

$$V_{t+1} = (1 - \zeta)V_t + \zeta\delta\delta', \tag{7}$$

where $\delta = Q_{t+1} - Q_t$ is the (scaled) temporal-difference error vector. For the standard tabular case, all elements of δ will be 0 apart from the element corresponding to the action a, which is the one to be updated and the covariance matrix $\delta\delta'$ will have a single non-zero diagonal element.

By re-arranging the terms of (7) we arrive at

$$V_{t+1} - V_t = \zeta(\delta\delta' - V_t) \tag{8}$$

which can be written in expanded form as

$$V_{t+1}(s,a) - V_t(s,a) = \zeta(\delta(s,a) - V_t(s,a)). \tag{9}$$

In the following we briefly describe how eligibility traces can be integrated in our framework.

Eligibility Traces and Variance Estimates Let us assume that the return R_t is given by a probability distribution of the form $p(R_t|s_t, a_t, \pi)$. Clearly, we may estimate $E[R_t|s_t, a_t, \pi]$ by averaging the returns while following policy π. However, we can assume that the distribution of R_t depends upon the distribution of R_{t+1}. We assume an exponential distribution for this prior dependency and thus we have $p(R_{t+1}|s_{t+1}, a_{t+1}, \pi) = \lambda p(R_{t+1}|s_t, a_t, \pi) + (1-\lambda)\mathcal{W}$, where $\mathcal{W}$ is the distribution of some unknown process.

The relation to eligibility traces is clear. We assume that an exponential prior in time governs the probability distribution of R_t. Thus, we can perform importance sampling on our parameters through the use of this prior: in other words each new sample should influence each parameter according to its importance weight.

Let us remind that in RL methods employing eligibility traces, the update δ is applied to all the evaluations of all state-action pairs (s, a) proportionally to the eligibility trace $e(s, a)$. By viewing eligibility traces as importance weights we can integrate them easily with our variance estimates. This results in the following update for each parameter's estimate.

$$V_{t+1}(s,a) = (1 - \zeta e(s,a))V_t(s,a) + \zeta e(s,a)\delta\delta', \tag{10}$$

or in compact form

$$V_{t+1} = (I - \zeta e)V_t + \zeta e\delta\delta', \tag{11}$$

where I is the identity matrix.

Function Approximation Methods We consider approaches where the value function is approximated with a parametrised function $Q_\theta : \mathcal{S} \to \mathbb{R}^{|\mathcal{A}|}$.

Gradient methods are a commonly used method for adapting the parameters θ. Given $\frac{\partial Q}{\partial \theta}\frac{\partial C}{\partial Q} \equiv \nabla_\theta Q \nabla_Q C$, we consider an update of the form $M_t = \theta_t + d_t$ for our parameters, where d_t is the gradient descent update. Then we simply apply (7) for this case and we obtain a covariance matrix for the parameters.

3 Conclusion

In this paper, we proposed a simple framework for action selection. that employs simple techniques for the estimation parameter distributions. The are nevertheless quite adequate for the task and which bypass the need for state-action visit counts. In preliminary experiments it was found that the use of the smoothed gradient estimate is particularly efficient in some tasks. On the other hand, the naive variance estimates that we outline are a generalisation of simple counting schemes and the scheme used in the prioritised sweeping algorithm [4] and that used in the RI method [5]. We feel that the connection between those estimates, the gradient, and its relation to convergence offer some justification to the previously ad hoc use of such techniques.

In preliminary experiments[3], we have used a naive sampling method for action selection, wherein the actions are selected proportionally to the probability of their being the best action. Future work would include investigating the use of explicit estimates for the value of exploration, which is one of the approaches outlined in [2]. Thare are also some interesting theoretical questions, such as the relationship of our model, and its possible application to policy-gradient methods (i.e. [1]).

References

1. Jonathan Baxter and Peter L. Bartlett. Reinforcement learning in POMDP's via direct gradient ascent. In *Proc. 17th International Conf. on Machine Learning*, pages 41–48. Morgan Kaufmann, San Francisco, CA, 2000.
2. Richard Dearden, Nir Friedman, and Stuart J. Russell. Bayesian Q-learning. In *AAAI/IAAI*, pages 761–768, 1998.
3. Christos Dimitrakakis and Samy Bengio. Estimates of parameter distributions for optimal action selection. Technical Report 04-72, IDIAP, 2004.
4. Andrew W. Moore and Christopher G. Atkeson. Prioritized sweeping: Reinforcement learning with less data and less time. *Machine Learning*, 13:103, 1993.
5. Yutaka Sakaguchi and Mitsuo Takano. Reliability of internal prediction/estimation and its application. i. adaptive action selection reflecting reliability of value function. *Neural Networks*, 2004. in press.
6. Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.