IDIAP RESEARCH REPORT

Estimates of Parameter Distributions for Optimal Action Selection

Christos Dimitrakakis [a] Samy Bengio [b]

IDIAP–RR 04-72

January 2005

Dalle Molle Institute for Perceptual Artificial Intelligence • P.O.Box 592 • Martigny • Valais • Switzerland

phone +41 − 27 − 721 77 11
fax +41 − 27 − 721 77 12
e-mail secretariat@idiap.ch
internet http://www.idiap.ch

[a] IDIAP, CP952, 1920 Martigny, Switzerland, dimitrak@idiap.ch
[b] IDIAP, CP952, 1920 Martigny, Switzerland, bengio@idiap.ch

IDIAP Research Report 04-72

Estimates of Parameter Distributions for Optimal Action Selection

Christos Dimitrakakis   Samy Bengio

January 2005

1 Introduction

Stochastic parameter estimation methods maintain an estimate θ_t of optimal parameters θ^* such that an optimality criterion C is maximised, i.e. where $\theta^* = \arg\max_\theta C(\theta)$. The optimisation itself takes place in such a way that $\theta_{t+1} = \mathcal{M}(C(\theta_t), \theta_t)$, where $\mathcal{M}$ is an operator that finds a parameter for which the optimality criterion is maximised (or merely increased). The series $\{\theta_t\}$ thus represents a trajectory in parameter space. Let us assume that for any subsequence starting with some θ_0, there exists a θ_∞ to which it converges[1]. However $\mathcal{M}$ itself, does not retain any memory with respect to the trajectory and thus at any time t the amount of convergence is unknown.

We may attempt to ameliorate this by maintaining an estimate of how close to convergence we are. This is not to be expressed as a measure with respect to the real parameters, which are anyway unknown, but as a measure derived from the estimation trajectory. By taking into account the stochasticity of the optimisation process, we can arrive at a probability distribution for the parameters. Thus, at any point in time, further to maintaining a current parameter estimate, we also have some knowledge about its distribution. This, in turn, is informative with respect to the confidence we have for each parameter, and of course for the estimation process as a whole. At the initial steps of an optimisation procedure the estimated distribution of parameters will be spread (indicating low confidence), while, as the process converges, it will become sharp (indicating high confidence).

For the case of value-based reinforcement learning, we use such a method to determine the distributions of the value of each possible action. These can be viewed succinctly as confidence measures for our action value estimates. In this respect, our results are similar to earlier ones, e.g. [5]. However our results are more general, in that they can be applied to any type of parametric model, and thus allow the representation of arbitrary action value distributions.

In the remainder of the article, we will describe our framework for the estimation of distributions of random parameters and its applications. In section 2 the distribution estimation framework is formulated. Section 3 outlines its application to estimation in probabilistic models. Section 4 is concerned with its application to value-based reinforcement learning and describes relationships with other models, especially in the particular case of linear and tabular action value methods. Section 5 discusses experimental results in a few simple reinforcement learning problems. We conclude with a discussion on the relationship between various methods and ours.

2 Parameter Distributions

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 parametrised 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 some settings (i.e. reinforcement learning), samples of f^* are generated actively. Asymptotic convergence results exist for such methods under the condition that samples are generated in a certain way, for example by assuming that each state-action pair is sampled infinitely often. In such settings it may be possible to generate samples in an optimal sense if we maintain a distribution of θ_t rather than a simple vector of parameters and we generate samples according to it. While this will not necessarily result in asymptotically better solutions, the use of such distributions can potentially improve the small-sample convergence of the policy to the optimal one.

First we define our method in a general setting. We start from viewing $\mathcal{M}$ as a stochastic process, from which we are sampling at every time step. We assume that the process is continuous with respect to the parameters at every point a (def.A.1). This is true for stochastic gradient methods, as follows

[1]For most estimators of interest, we have $E[\theta_\infty] = \theta^*$. For the more restricted class of unbiased estimators, we have $E[\theta_t] = \theta^* \forall t > 0$

from the fact that $\|e_t\| \leq \delta$ (see definition A.4). In the following we will use the continuity assumption to impose an exponential prior on the distribution of $\mathcal{M}$ with respect to θ.

2.1 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. One such measure is given by the sample variance $(E[M_t] - \theta_{t+1})(E[M_t] - \theta_{t+1})'$.

Of course it is not obvious what the expected value of M_t is. Firstly, we assume that it is bounded [2] 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 $\hat{E}[M_t] = \theta_t$.

A slightly more sophisticated method assumes that $\mathcal{M}$ is Lipschitz continuous[3] with respect to θ, which leads to

$$\hat{E}[M_t] = (1 - \eta)\hat{E}[M_{t-1}] + \eta(\theta_t - \theta_{t-1}), \tag{1}$$

where we are making use of an exponential prior for the distribution of estimator moments.

Using such a prior, we obtain a variance estimate of the form

$$V_{t+1} = (1 - \zeta)V_t + \zeta(\hat{E}[M_t] - \theta_{t+1})(\hat{E}[M_t] - \theta_{t+1})'. \tag{2}$$

where we use V_t for our estimate of the variance of $\mathcal{M}(C, \theta_t)$. We may plug in either (1) or θ_t for $E[\hat{M}_t]$, or indeed any other mean estimate. We discuss the resulting expressions in the two simplest cases.

Definition 2.1 (Naive variance estimate) *By assuming that $\mathcal{M}$ is a zero-mean process, i.e. that $E[M_t] = \theta_t$, we have:*

$$V_{t+1} = (1 - \zeta)V_t + \zeta(\theta_t - \theta_{t+1})(\theta_t - \theta_{t+1})'. \tag{3}$$

Definition 2.2 (Counting variance estimate) *By assuming $E[M_t] = \theta_{t+1}$, i.e. that $\mathcal{M}$ is a deterministic process, we have:*

$$V_{t+1} = (1 - \zeta)V_t. \tag{4}$$

The latter method is equivalent to a class of counting schemes. With an appropriate choice for ζ such schemes can be adequate for some problems. Note that in the case where we maintain a set of parameters which are updated separately (such as in tabular reinforcement learning methods), then it is appropriate to maintain separate variance estimates of this type.

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.

2.1.1 Relation of Estimates to Convergence

In the general case, estimating $|\theta - \theta^*|$, the distance to a solution, can be as difficult as determining θ^* itself.[4] However for the variance estimates given above we can perform a simple analysis that indicates its relation to convergence as follows: Let us assume two sequences $\{a\}$ and $\{b\}$ such that $\lim_{t\to\infty} a_t = a_\infty$, $\lim_{t\to\infty} b_t = b_\infty$. If they converge to the same p, then, for each t there exists δ_a and δ_b such that for all $k > t$, $\|a_k - p\| < \delta_a$ and $\|b_k - p\| < \delta_b$. If without loss of generality $\delta_a < \delta_b$ then

[2] 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.

[3] This means that M_t will tend to move in the same general direction that previous operators did, i.e. that $E[M_t]$ depends upon $E[M_{t-1}]$. This follows from the boundedness assumption.

[4] If we can estimate the distance from the solution in a Euclidean space, this is sufficient to actually find the solution with complexity linear in the dimensionality of the parameter space.

it holds that both $\|b_k - p\|$ and $\|a_k - p\|$ are bounded by δ_b. For the case where there exists a set A of local minima of the cost function such that A is a limit set $\forall\ \theta$, then we have that

$$\begin{aligned} E^2[\theta_t - \theta_{t-1}] \leq E^2[\theta_t] \\ \leq E^2[a - b | a \in A_1, b \in A_2, A_1 \cap A_2 = \emptyset] \end{aligned} \tag{5}$$

Typically $V_t \geq E^2[\theta_t]$, as it is a biased estimator of variance, but we assume that there exists $\epsilon > 0$ such that $V_t \leq E^2[\theta_t] + \epsilon$. This allows us to place loose bounds on our estimates.

$$E^2[\theta_t] \leq V_t - \epsilon \leq E^2[a - b | a \in A_1, b \in A_2, A_1 \cap A_2 = \emptyset] \tag{6}$$

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 simple to show that the mean update we have defined is an approximate measure of the gradient under some conditions.

From (1), we have

$$\begin{aligned} \hat{E}[M_{t+1}] = \sum_{k=1}^{N} (1 - \eta_k)\hat{E}[M_k] + \eta_k \alpha(\delta_k + e_k) \\ = (1 - \eta_k)^t \hat{E}[M_1] + \sum_{k=1}^{t} (1 - \eta_k)^{t-k} \eta_k \alpha(\delta_k + e_k) \\ = (1 - \eta_k)^t \hat{E}[M_1] + \eta_k \alpha(\sum_{k=1}^{t} (1 - \eta_k)^{t-k} \delta_k + \sum_{k=1}^{t} (1 - \eta_k)^{t-k} e_k) \end{aligned} \tag{7}$$

For the case when $\eta_k = 1/k$ we have,

$$\lim_{\alpha \to 0} \hat{E}[M_t] - \theta \approx \nabla C(\theta) \tag{8}$$

with better approximation as $N \to \infty$. We also note that our variance estimate is similarly related to the gradient, via

$$\mathrm{trace}(V) \approx \|\nabla C(\theta)\|^2 + E^2[\epsilon],$$

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

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 (see A.1 for a proof):

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

$$m\|z\|^2 \leq z' \nabla^2 C(\theta) z, \quad \forall\ z \in \mathbb{R}^n, \tag{9}$$

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

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

Thus, both 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.

3 Estimation and Sampling

We are concerned with the problem of estimating the probability of an event given some data. Let us assume a function $f(x, \theta)$ representing this probability, where x and θ are data and parameters respectively[5]. If θ is probabilistic and $f(x, \cdot)$ is non-linear, then the calculation of this probability does not have a closed form. We instead draw a set of values $\{\theta_1, \ldots, \theta_n\}$ from $p(\theta)$ and then we approximate the distribution of f by

$$f(x) = \int f(x|\theta)p(\theta)d\theta \approx \frac{1}{n}\sum_{i=1}^{n} f(x|\theta_i). \tag{10}$$

This is the basic premise of Monte Carlo sampling.[6] Of course, if θ is drawn from some other distribution than $p(\theta)$ then we need to apply techniques such as importance sampling in order to weigh the relation of each drawn sample to the target distribution. Although we shall not be making use of importance sampling techniques in the sequel, we will draw some parallels between it and reinforcement learning.

4 Optimal Action Selection

Most, if not all, reinforcement learning 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

$$\pi^* = \arg\max_{\pi} E[R_t|\pi], \tag{11}$$

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

An in important subset of reinforcement learning methods is formed by value-based methods (these 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 $\delta = r_t + \gamma\hat{E}[R_{t+1}|s_{t+1} = u, a_t = b, \pi] - Q(s, a)$. 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.

Research on action-value methods has naturally concentrated on the estimation of action values. Because action values are also used to determine the current policy, such methods suffer from the problem that without sufficient exploration convergence can be slow. Off-policy methods, such as Q-learning algorithm [7], try to approximate the value function under the optimal policy while following an arbitrary one and thus alleviate the problem to some extend. In general, however, most methods focus on the problem of approximating the value function, while the action selection mechanism tends to be one of the following:

Definition 4.1 (ϵ-greedy) *With probability $\epsilon > 0$, a random action $a \in \mathcal{A}$ is selected. Otherwise $\arg\max_a Q(s, a)$ is selected.*

[5] In general there is no need for distinctions between data and parameters. A probability distribution must be assumed for both, even if it is a singular one.

[6] See [3] for an overview of current methods and applications.

Definition 4.2 (Soft-max) *In this method, actions are sampled according to a Gibbs distribution*

$$p(a|s, Q) = \frac{e^{Q(s,a)/\tau}}{\sum_{b \in A} e^{Q(s,b)/\tau}}, \tag{12}$$

where $\tau > 0$ is referred to as the temperature.

Both ϵ and τ in the above examples can be viewed as confidences we have and represent our policy for sampling the space. However such methods require significant tuning in order to apply reinforcement learning techniques to a problem. In particular, pessimistic initial values for Q tend to slow the convergence of algorithms employing the above action selection techniques, especially when returns are stochastic. Some of these issues have been recently addressed to some extent in [5], in which the following action selection method was proposed

Definition 4.3 (Reliability Index) *In this method, actions are sampled according to a Gibbs distribution*

$$p(a|s, Q) = \frac{e^{\eta Q(s,a)/\sqrt{v_s}}}{\sum_{b \in A} e^{\eta Q(s,b)/\sqrt{v_s}}}, \tag{13}$$

where $R_s > 0$ is defined $\forall\ s \in \mathcal{S}$ and is a variance estimate *for each one of our Q estimates and η is a free parameter.*

In the sequel we will apply our variance estimates to the reinforcement learning problem and describe appropriate action selection mechanism that arise from our framework. This enables us to form a natural method for the estimation of action probabilities in order to achieve a good small-sample convergence.

4.1 Application of Variance Estimates to Action Values

By applying equation (2) under the assumption that $E[M_t] = \theta_t$, we obtain

$$V_{t+1} = (1 - \zeta)V_t + \zeta(M_t - \theta_{t+1})(M_t - \theta_{t+1})'. \tag{14}$$

For the remainder we shall be using this estimate, although an estimate derived from (1) could have also been applied. However this suffices for our current purposes and it also shows some parallels with previous methods developed for the tabular case.

In the following short sections we consider the application of such estimates 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.

4.1.1 Alternative Approaches

There have been previous applications of such methods to the problem of action selection in reinforcement learning. In [2], the authors take a Bayesian approach for estimating parameter distributions, by defining a normal-gamma prior distribution for the parameters to be estimated, whose posterior distribution after a set of observations remains a normal-gamma distribution. They also consider two different action selection schemes: (a)Q-value sampling, in which the probability that the return of one action is larger than those of others is approximately calculated and (b)Myopic-VPI selection, in which the value to be gained by exploring actions is estimated. This method has only recently come to our attention and thus we have not performed yet performed an experimental comparison with this method and ours.

The second approach that we are currently aware of is the Reliability Index method, described in [5]. This method has substantial similarity to our own for tabular action value methods using the parameter variance update given by (3). This particular case is discussed in the following section.

4.1.2 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 (14), we obtain

$$V_{t+1} = (1-\zeta)V_t + \zeta\delta\delta', \tag{15}$$

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 (15) we arrive at

$$V_{t+1} - V_t = \zeta(\delta\delta' - V_t) \tag{16}$$

which can be written in expanded form as

$$V_{t+1}(s,a) - V_t(s,a) = \zeta(\delta(s,a) - V_t(s,a)) \tag{17}$$

As this is similar to the case of [5], we pause for a moment to ponder the differences. Firstly, the authors there consider a form of the type $V_{t+1}(s) - V_t(s) = \zeta(\delta(s) + \gamma V_t(s') - V_t(s))$, with a common V for all actions, or of a type $V_{t+1}(s,a) - V_t(s,a) = \zeta(\delta(s) + \gamma V_t(s',a) - V_t(s,a))$, but they then average over the states to obtain $V_t(s) = \frac{1}{|\mathcal{A}|}\sum_a V_t(s,a)$. In either case they select actions with (13). Secondly, they have used a temporal-difference type of update, since $V_t(s')$ is the estimated variance of next state's evaluation. The authors postulate that this represents the dependency between the reliability of one state and the ones close to it. In our view, parameter variances are directly related to parameter updates and γ is related to the utility function rather than to assumptions about parameter dependencies. However, a model for setting priors about parameter dependencies is offered by the exponential prior, commonly used in eligibility traces. In the following we briefly describe how eligibility traces can be integrated in our framework.

4.1.3 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 $p(R_{t+1}|s_{t+1}, a_{t+1}, \pi) = \lambda p(R_{t+1}|s_t, a_t, \pi) + (1-\lambda)\mathcal{N}$, where $\mathcal{N}$ 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. 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.

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.

$$V_{t+1}(s,a) = (1 - \zeta e(s,a))V_t(s,a) + \zeta e(s,a)\delta\delta', \tag{18}$$

or in compact form

$$V_{t+1} = (I - \zeta e)V_t + \zeta e\delta\delta', \tag{19}$$

where I is the identity matrix.

4.1.4 Function Approximation Methods

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

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 (15) for this case and we obtain a covariance matrix for the parameters.

4.2 A Unified View of Action Selection

We are proposing two methods for using our variance estimates. The first is a natural action selection mechanism for the case of linear approximations (including the tabular case), since the variance of Q depends linearly on those.

4.2.1 Sampling Actions from Action Distributions

Consider the probability of action a being the optimal action for some state s. We write this as $P(a = a^*|s)$. We need to obtain the posterior distribution of this for all actions, given the distribution of Q and the state, [7] denoted $P(a = a^*|Q, s)$. The Bayes rule gives

$$P(a = a^*|Q, s) = \frac{p(Q|a, s)p(a = a^*|s)}{\sum_{b \in \mathcal{A}} P(Q|b, s)P(b = a^*|s)}, \tag{20}$$

where we have made use of the fact that $\sum_{b \in \mathcal{A}} P(b = a^*|s) = 1$. Now we must assume a distribution family for $p(Q|a, s)$. We consider the Boltzmann distribution which can be written as

$$p(E|i) = e^{-E_i/K\tau}$$

and has a physical interpretation of the distribution of the energies E of particles in a given state i. We will be using this in the following to define a soft-max method for selection actions:

Definition 4.4 (Bayes-greedy) *Select actions a according to probabilities:*

$$P(a = a^*|Q, s) = \frac{\exp(Q(s, a)/\sqrt{v_{s,a}})}{\sum_b \exp(Q(s, b)/\sqrt{v_{s,b}})} \tag{21}$$

For the tabular case, $v_{s,a}$ at time t is simply $V_t(s, a)$. For the linear case, in which $Q(s, a) = \sum_i w_{i,a} s_i$, where w are components of a weight matrix and s_i is the i-th component of the state vector, the variance is simply $v_{s,a} = \sum_i w_{i,a} V_t i, a$ where $V_t(i, a)$ is the variance of the weight $w_{i,a}$. Of course we could also consider a full covariance matrix.

4.2.2 Sampling Actions via Parameter Sampling

The second method applies to the more general function approximation case. Here we have to choose a distribution for our parameters; and then we sample from this distribution to generate actions, rather than postulating a distribution for Q and sampling from that. This is because in the general case it is difficult to determine the distribution of Q from that of the parameters. However, sampling from the parameter distribution directly is the same as sampling from the Q distribution.

[7] In our model Q is no longer a single value but a distribution characterised by the variance V. In this section we make no distinction between our estimate of the mean and the actual distribution in order to minimise symbol use.

Definition 4.5 (Sampling-Greedy) *In this method, action sampling arises from sampling in the parameter space. At each time, we select action a^* such that*

$$a^* = \arg\max_a Q(s, a, \Theta), \tag{22}$$

where $\Theta = \mathcal{N}(\theta, V_t)$ is a sample randomly drawn from a distribution with mean θ and variance V_t.

It is possible to refine this method by sampling multiple times and keeping the action that was selected the most. However we feel that for reinforcement learning this is not as necessary, as we do not need to estimate the distribution, but merely to sample from it in the first place. This means that it is possible to have arbitrary approximation architectures and still be able to perform optimally greedy sampling.

5 Experiments

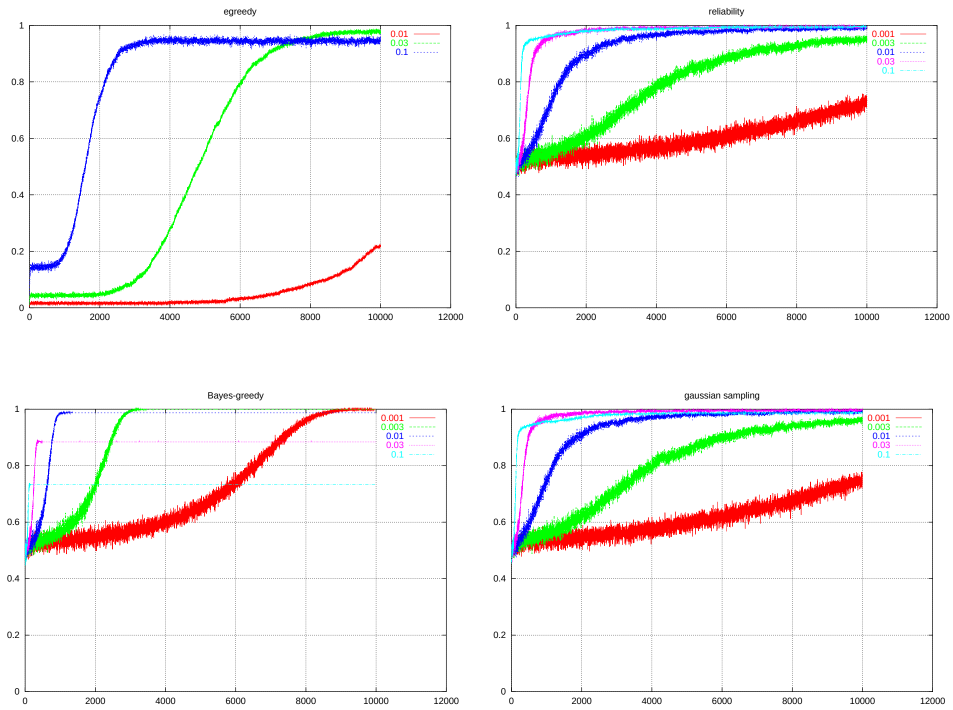

Figure 1: The above figures display results for the 2-armed bandit problem for the ϵ-greedy, reliability index, and our Bayes-greedy sampling-greedy methods respectively.

In this paper we only present preliminary results on n-armed bandit problems. Although our method is applicable to other tasks and is compatible with eligibility traces, we find that bandit tasks represent a problem that our method should be able to deal with effectively and which are easy to analyse. It also provides a framework within which some of the differences between our method and the RI are eliminated and it banishes the need for making a comparison between using off-policy or on-policy estimates. However in future work we will be applying our method to tasks with state.

In the n-armed bandit task employed herein, the environment is composed of a single state and n actions. Each action a has a reward of 1 with probability p_a and a reward of 0 with probability

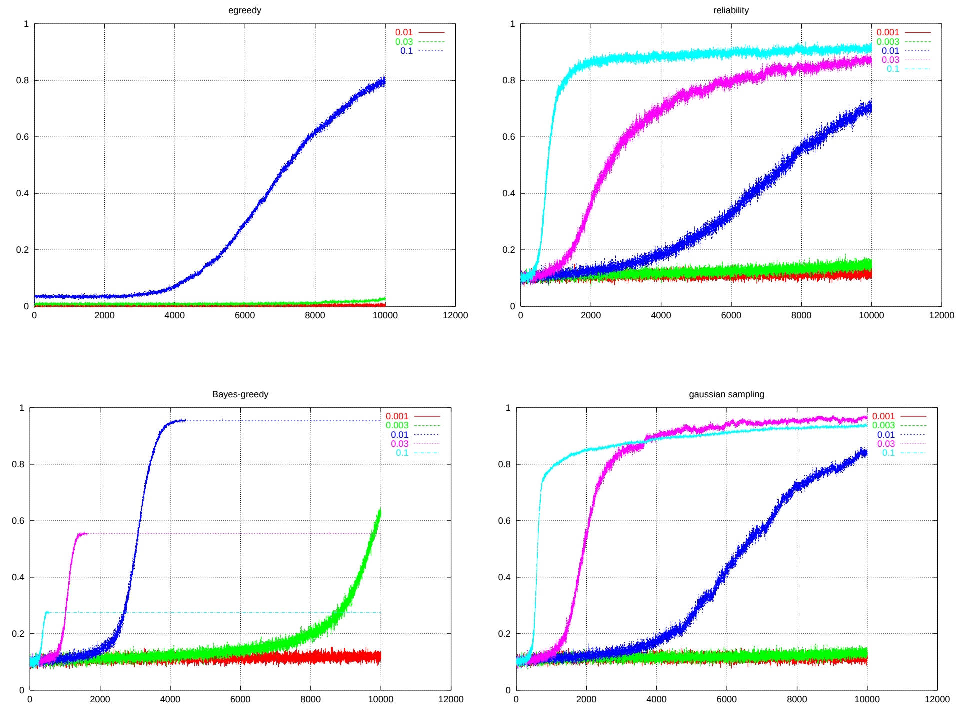

Figure 2: The above figures display results for the 10-armed bandit problem for the ϵ-greedy, reliability index, and our Bayes-greedy sampling-greedy methods respectively.

$1 - p_a$. For all actions apart from one, a^*, this probability is randomly generated at the start of each experiment in the range $[0, 0.5]$. p_{a^*} is set to 0.6. We perform a comparison of various action selection methods by running 1000 experiments with a randomly generated set of p_a each time. Each experiments runs for 10000 iterations. The plots show the average number of times the optimal action was selected at each iteration. We ran experiments with 2, 10, 20 and 100 arms. Figures 5, 5 and 5 show results with 2, 10 and 20 arms respectively. The legend refers to the parameter ζ for our methods and the reliability index method and to ϵ for the ϵ-greedy method.

The parameters of the various algorithms were set up as follows: For ϵ-greedy action selection, ϵ was in the set $\{0.01, 0.03, 0.1\}$. For soft-max action selection, the temperature parameter τ was set to $\{0.01, 0.1, 1\}$. For reliability index and our methods, the ζ was in $\{0.01, 0.03, 0.1\}$. The learning rate was set to 0.1 for all methods and γ was set to 0 since there is only a single state. In this case our update for $V_t(s, a)$ is the same for both our method and the reliability index, so in order to estimate the v_s employed by the reliability index, we used our estimates to obtain $v_s = \sqrt{\frac{1}{|\mathcal{A}|} \sum_{a \in \mathcal{A}} V_t(s, a)}$ and constrain $V_t(s, a)$ to a minimum value 0.0001. It was bounded by the same value for our method.

Results show that our method, after an initial exploratory period, converges much faster and more often than other action selection methods. The percentage of runs that were performing optimally is close to 99% for the 2 and 10-arm problems and the RI method was performing similar to the gaussian parameter sampling. For the 20-arm problem the gaussian parameter sampling was clearly better. For the 100-arm problems, we achieved a 50% accuracy, while other methods failed to go above 10% after 10000 iterations. Results for the soft-max method are similar to that of ϵ-greedy: they exhibit a high sensitivity.

We would further wish to note the choice of convergence estimates influences the algorithmic

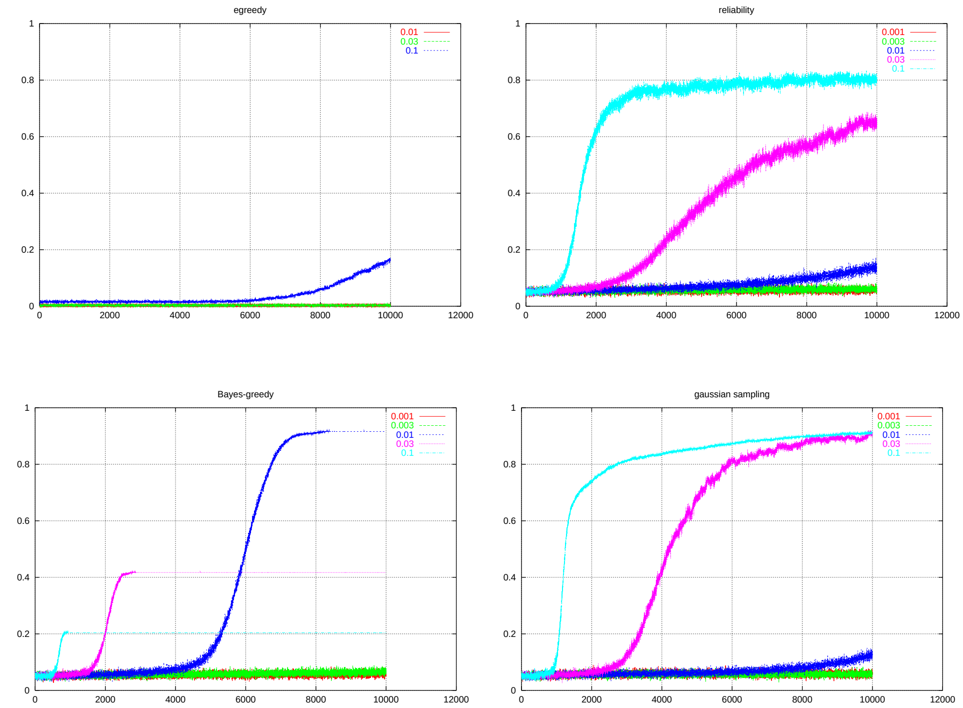

Figure 3: The above figures display results for the 20-armed bandit problem for the ϵ-greedy, reliability index, and our Bayes-greedy sampling-greedy methods respectively.

behaviour significantly in some cases. For the bandit problem discussed here, using the variance update 4, results in better performance for the action sampling method, while the parameter sampling method and the reliability index methods perform slightly worse than before.

6 Conclusion

We presented a general probabilistic framework for action selection. It employs simple techniques for the estimation parameter distributions, which are nevertheless quite adequate for the task and which bypass the need for state-action visit counts. We furthermore integrate our parameter distributions with reinforcement learning with eligibility traces. The action selection distribution can be derived from the parameter distributions analytically for the linear case. This results in an optimally greedy action sampling. For the general case, a distribution over actions cannot be formulated analytically, so we have defined a simple sampling mechanism to sample over the actions themselves.

Both the action sampling method and the parameter sampling method seem to perform well, but they exhibit a qualitative difference. The reason for this might be that not enough parameter samples were used to generate action samples. Having shown that the method works in principle in these simple problems, in future work we plan to perform experiments in more demanding tasks. Since our method can also be applied in the case of function approximation and eligibility traces, we plan to test these cases. Other future work includes investigating different types of estimates, establishing a relationship between our method and the Bayesian approach if possible and finally, using

Lastly, we have not touched some interesting theoretical questions, such as the relationship of our model, and its possible application, to policy-gradient methods (i.e. [1]). Another possible direction is to consider other forms for our analytically defined action sampling, such that it is greedy with respect to a *different* criterion than the probability of each action being the optimal one.

Lastly, we do not have any small-sample convergence results for our framework. One recent such result, obtained for the n-armed bandit problem was presented in [4]. We aim to start working in this direction in the near future, in the hope of achieving a better understanding of the necessary conditions for quick convergence.

Acknowledgements

This work was supported in part by the IST Program of the European Community, under the PASCAL Network of Excellence, IST-2002-506778, funded in part by the Swiss Federal Office for Education and Science (OFES) and the Swiss NSF through the NCCR on IM2.

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] Arnaud Doucet, Nando de Freitas, and Neil Gordon, editors. *Sequential Monte Carlo Methods in Practive*. Springer, 2001.

[4] Shiee Mannor and John N. Tsitsiklis. The sample complexity of exploration in the multi-armed bandit problem. *Journal of Machine Learning Research*, 5:623–648, 2004.

[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.

[7] Christopher J.C.H. Watkins and Peter Dayan. Technical note Q-learning. *Machine Learning*, 8:279, 1992.

A DEFINITIONS AND NOTATION

The ℓ_p, norms are denoted by $\|\cdot\|_p$ while generally we will drop the subscript for the ℓ_2 norm and we will note the ℓ_1 norm with $|\cdot|$. The dimensionality of a space $\mathcal{S}$ is denoted by $|\mathcal{S}\|$. Probabilities and probability density functions with the shorthand notation $p(\cdot)$ whenever there is no room for ambiguity. The expectation and variance operators are noted by $E[\cdot]$ and $E^2[\cdot]$ respectively. Estimated values, such as the sample mean, will be noted in the form $\hat{E}[\cdot]$.

Definition A.1 (Point continuity) *A function f is continuous at a if $\forall\ \epsilon$ there exists a δ such that*

$$\|f(a) - f(b)\| < \epsilon$$

for all b such that $\|a - b\| < \delta$.

Definition A.2 (Lipschitz Continuity) *A function f is Lipschitz continuous on R if, for some L,*

$$\|\nabla f(a) - \nabla f(b)\| \leq L\|a - b\|$$

holds for all $a, b \in R$.

Definition A.3 (Gradient method) *A gradient method for minimising a cost function C generates a sequence $\{\theta\}$ according to*

$$\theta_{t+1} = \theta_t + \alpha_t d_t,$$

with $\alpha_t > 0$ and where d_t is such that

$$\nabla C(\theta_t)' d_t < 0$$

if $\nabla C(\theta_t) \neq 0$,

Definition A.4 (Stochastic gradient method) *A stochastic gradient method generates a sequence $\{\theta\}$ for minimising C according to*

$$\theta_{t+1} = \theta_t + \alpha_t(d_t + e_t),$$

with $\alpha_t > 0$, where d_t is such that

$$\nabla C(\theta_t)' d_t < 0,$$

if $\nabla C(\theta_t) \neq 0$, and where e_t is the outcome of a random process with the property $\|e_t\| \leq \delta$ for all t.

Definition A.5 (Subgradient) *Given a function $f : \mathbb{R}^n \to \mathbb{R}$, that is convex in $A \subseteq \mathbb{R}^n$ a vector $d \in A$ is a subgradient of f at point $x \in A$ if*

$$f(y) \geq f(x) + (z - x)'d, \quad \forall z \in A.$$

For the case of f being concave in A, d is a subgradient of f at x if $-d$ is a subgradient of $-f$ at x. The set of all subgradients of f at x is called the subdifferential *of f at x and is denoted by $\partial f(x)$.*

Definition A.6 (Limit sets) *A sequence $\{\theta\}$ converges to a set $A \subset B$ if, $\forall\ \theta_0 \in B$, $\exists\ k$ such that $\theta_t \in A\ \forall\ t \geq k$. Furthermore consider the sequence of sets $\{A\}$, for increasing values of k. If the sequence convergences, say to A_l, then A_l is the* limit set *of the sequence $\{\theta\}$.*

Theorem A.1 (Markov inequality)

$$P(X \leq a) \leq E[X]/a \forall X \geq 0 \tag{23}$$

Theorem A.2 (Chebyshev inequality) *The Chebyshev inequality follows from the Markov inequality.*

$$P(|X - m| \leq a) \leq \sigma^2/a^2 \tag{24}$$

Theorem A.3 (Chernoff bound)

$$P(|X| \leq \lambda\sigma) \leq 2e^{-\lambda^2/4} \tag{25}$$

A.1 Distance Bound

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

$$m\|z\|^2 \leq z'\nabla^2 C(\theta)z, \quad \forall\ z \in \mathbb{R}^n, \tag{26}$$

then, for all $\theta \in S$,

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

For any twice continously differentiable function f, it holds that:

$$\nabla f(y) = \nabla f(x) + \int_0^1 \nabla^2 f\big(x + t(y-x)\big)(y-x)dt.$$

We apply this to C and note that $\nabla C(\theta^*) = 0$ to obtain:

$$\nabla C(\theta) = \int_0^1 \nabla^2 C\big(\theta^* + t(\theta - \theta^*)\big)(\theta - \theta^*)dt$$

$$(\theta - \theta^*)'\nabla C(\theta) = \int_0^1 (\theta - \theta^*)'\nabla^2 C\big(\theta^* + t(\theta - \theta^*)\big)(\theta - \theta^*)dt.$$

From (26), we have:

$$\begin{aligned}(\theta - \theta^*)'\nabla C(\theta) &\geq m\|\theta - \theta^*\|^2 \\ \|\theta - \theta^*\|\|\nabla C(\theta)\| &\geq m\|\theta - \theta^*\|^2 \\ m\|\theta - \theta^*\|^2 &\leq \|\nabla C(\theta)\|/m.\end{aligned}$$

The second statement can be proven by using the following second order expansion that holds for every function f that is twice continuously differentiable over an open sphere f centred at x, and with $y : x + y \in S$:

$$f(x+y) = f(x) + y'\nabla f(x) + \frac{1}{2}y'\nabla^2 f(x)y + o(\|y\|^2) \tag{27}$$

from which it follows that:

$$f(y) - f(x) = f(x + (y-x)) - f(x) = (y-x)'\nabla f(x) + \frac{1}{2}(y-x)'\nabla^2 f(x)(y-x) + o(\|y-x\|^2) \tag{28}$$

We also need the fact that

$$\min_{y\in\mathbb{R}^n} \big\{(y-x)'\nabla f(x) + m\|y-x\|^2/2\big\} = -\frac{1}{2m}\|\nabla f(x)\|^2. \tag{29}$$

(This can be proven by the fact that at the minimum, the derivative of the argument of the minimum operator will have a derivative of 0, resulting in $y^* = \frac{-\nabla f(x)}{m} + x$. A substitution completes the proof.)

From (26) and (28), we have:

$$\begin{aligned}f(x + (y-x)) - f(x) &= (y-x)'\nabla f(x) + \frac{1}{2}(y-x)'\nabla^2 f(x)(y-x) + o(\|y-x\|^2) \\ &\geq (y-x)'\nabla f(x) + \frac{m}{2}\|y-x\|^2\end{aligned}$$

We can then replace the right hand side with its minimum, as given by (29), which gives:

$$f(x + (y-x)) - f(x) \geq -\frac{1}{m}\|\nabla f(x)\|^2.$$

We now replace

$$C(\theta) - C(\theta^*) \leq \frac{1}{2m}\|\nabla C(\theta)\|^2.$$

We further note that if $\|C(\theta)\| \leq \epsilon$ then the following inequalities also hold:

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