in: Proc. Int. Conf. on Artificial Neural Networks (ICANN 2009), Cyprus,

Part I, LNCS 5768, pp. 70-79, Springer Verlag 2009

Efficient Uncertainty Propagation for Reinforcement Learning with Limited Data

Alexander Hans[1,2] and Steffen Udluft[1] [*]

1- Siemens AG, Corporate Technology, Information & Communications,
Learning Systems, Otto-Hahn-Ring 6, D-81739 Munich, Germany
{alexander.hans.ext|steffen.udluft}@siemens.com

2- Ilmenau Technical University, Neuroinformatics and Cognitive Robotics Lab,
P.O.Box 100565, D-98684 Ilmenau, Germany

Abstract. In a typical reinforcement learning (RL) setting details of the environment are not given explicitly but have to be estimated from observations. Most RL approaches only optimize the expected value. However, if the number of observations is limited considering expected values only can lead to false conclusions. Instead, it is crucial to also account for the estimator's uncertainties. In this paper, we present a method to incorporate those uncertainties and propagate them to the conclusions. By being only approximate, the method is computationally feasible. Furthermore, we describe a Bayesian approach to design the estimators. Our experiments show that the method considerably increases the robustness of the derived policies compared to the standard approach.

Key words: reinforcement learning, model-based, uncertainty, Bayesian modeling

1 Introduction

In reinforcement learning (RL) [12] one is concerned with finding a policy, i.e., a mapping from states to actions, that moves an agent optimally in an environment assumed to be a Markov decision process (MDP) $M := (S, A, P, R)$ with a state space S, a set of possible actions A, the system dynamics, defined as probability distribution $P : S \times A \times S \to [0, 1]$, which gives the probability of reaching state s' by executing action a in state s, and a reward function $R : S \times A \times S \to \mathbb{R}$, which determines the reward for a given transition. If the parameters of the MDP are known a priori, an optimal policy can be determined, e.g., using dynamic programming. Often, however, the MDP's parameters are not known in advance. A common way of handling this situation is model-based RL, where one first estimates a model of the MDP from a number of observations and then finds an optimal policy w.r.t. that model. In general, such a policy will not be optimal w.r.t. the real MDP. Especially in case of a limited number of observations the estimated MDP has a high probability to differ from the real one substantially. In this case, it is in particular possible to derive a policy that will perform badly when applied to the real MDP.

[*] The authors would like to thank the reviewers for very useful comments.

By incorporating the model estimators' uncertainties into the determination of the policy it is possible to weaken this problem. In recent work by Schneegass et al. [10] uncertainty propagation (UP) was applied to the Bellman iteration to determine the Q-function's [12] uncertainty and derive uncertainty incorporating policies. While the algorithm described in [10] provides significant advantages over methods not considering uncertainty, it adds a huge computational burden for updating the covariance matrix in each iteration. In this paper, we propose an algorithm called the *diagonal approximation of uncertainty incorporating policy iteration* (DUIPI) for discrete MDPs that represents an efficient way of using UP to incorporate the model's uncertainty into the derived policy by only considering the diagonal of the covariance matrix. Only considering the diagonal neglects the correlations between the state-action pairs, which in fact are small for many RL problems, where on average different state-action pairs share only little probabilities to reach the same successor states. DUIPI is easier to implement and, most importantly, lies in the same complexity class as the standard Bellman iteration and is therefore computationally much cheaper than the method considering the full covariance matrix. Although some of the results obtained with DUIPI are not as good as those of the full-matrix method, the robustness of the resulting policies is increased considerably, compared to the standard Bellman iteration, which does not regard the uncertainty. In this context it furthermore is advisable to use Bayesian statistics to model the *a posteriori* distributions of the transition probabilities and rewards in order to access the estimators' uncertainties properly. Additionally, it allows the specification of prior knowledge and the user's belief.

There have already been a number of contributions that consider uncertainties when estimating MDPs. E.g., the framework of robust MDPs has widely been studied (e.g., [8, 1]), in which one assumes that all uncertainties can only lie within a bounded set. One tries to find policies optimizing the worst case within that set, which often results in too conservative policies. Within the context of Bayesian RL, incorporation of prior knowledge about confidence and uncertainty directly into the approached policy is possible. E.g., Engel et al. applied Gaussian processes for policy evaluation by updating a prior distribution over value functions to posteriors by observing samples from the MDP [4, 5]. Ghavamzadeh and Engel presented additional Bayesian approaches to model-free RL [6, 7]. Using Gaussian processes inherently introduces a measure of uncertainty based on the number of samples. When dealing with model-based approaches, however, one starts with a natural local measure of the uncertainty of the transition probabilities and the rewards. In that context, related to the present paper is work by Delage and Mannor [3], who used convex optimization to solve the percentile problem and applied it to the exploration-exploitation trade-off. Model-based interval estimation (MBIE) was also used for efficient exploration by using local uncertainty to derive optimistic exploration policies, e.g., [11].

The remainder of the paper is organized as follows. In sec. 2 we describe how to incorporate knowledge of uncertainty into the Bellman iteration using

UP, sec. 3 presents ways of parameter estimation. Experiments and results are presented in sec. 4. Sec. 5 finishes the paper with a short conclusion.

2 Incorporation of Uncertainty

Our notion of uncertainty is concerned with the uncertainty that stems from the ignorance of the exact properties of the real MDP, as they are usually unknown and must be estimated from observations. With an increasing number of observations the uncertainty decreases; in the limit of an infinite number of observations of every possible transition the uncertainty vanishes as the true properties of the MDP are revealed. For a given number of observations the uncertainty depends on the inherent stochasticity of the MDP; if the MDP is known to be completely deterministic, one observation of a transition is sufficient to determine all properties of that transition; the more the MDP is stochastic, the more uncertainty will remain for a fixed number of observations. It is important to distinguish this uncertainty from an MDP's inherent stochasticity.

We want to use the knowledge of uncertainty to determine an optimal Q-function Q^* with its uncertainty σQ^*. In a second step it is then possible to change the Bellman iteration to not only regard a Q-value but also its uncertainty, resulting in a policy that generally prefers actions that have a low probability of leading to an inferior long-term reward.

2.1 Determining the Q-Function's Uncertainty

To obtain the Q-function's uncertainty, we use the concept of uncertainty propagation (UP), also known as Gaussian error propagation (e.g., [2]), to propagate the uncertainties of the measurements, i.e., the transition probabilities and the rewards, to the conclusions, i.e., the Q-function and policy. The uncertainty of values $f(x)$ with $f : \mathbb{R}^m \to \mathbb{R}^n$ is determined as $(\sigma f)^2 = \sum_i \left(\frac{\partial f}{\partial x_i}\right)^2 (\sigma x_i)^2$. The update step of the Bellman iteration,

$$Q^m(s,a) := \sum_{s'} \hat{P}(s'|s,a) \left[\hat{R}(s,a,s') + \gamma V^{m-1}(s')\right], \tag{1}$$

can be regarded as a function of the estimated transition probabilities $\hat{P}$ and rewards $\hat{R}$, and the Q-function of the previous iteration Q^{m-1} (V^{m-1} is a subset of Q^{m-1}), that yields the updated Q-function Q^m. Applying UP to the Bellman iteration, one obtains an update equation for the Q-function's uncertainty:

$$\begin{aligned}(\sigma Q^m(s,a))^2 := {} & \sum_{s'} (D_{QQ})^2 (\sigma V^{m-1}(s'))^2 + \sum_{s'} (D_{QP})^2 (\sigma \hat{P}(s'|s,a))^2 + \\ & \sum_{s'} (D_{QR})^2 (\sigma \hat{R}(s,a,s'))^2,\end{aligned} \tag{2}$$

$$D_{QQ} = \gamma \hat{P}(s'|s,a),\ D_{QP} = \hat{R}(s,a,s') + \gamma V^{m-1}(s'),\ D_{QR} = \hat{P}(s'|s,a).$$

V^m and σV^m have to be set depending on the desired type of the policy (stochastic or deterministic) and whether policy evaluation or policy iteration is performed. E.g., for policy evaluation of a stochastic policy π

$$V^m(s) = \sum_a \pi(a|s) Q^m(s,a), \tag{3}$$

$$(\sigma V^m(s))^2 = \sum_a \pi(a|s)^2 (\sigma Q^m(s,a))^2. \tag{4}$$

For policy iteration, according to the Bellman optimality equation and resulting in the Q-function Q^* of an optimal policy, $V^m(s) = \max_a Q^m(s,a)$ and $(\sigma V^m(s))^2 = (\sigma Q^m(s, \arg\max_a Q^m(s,a)))^2$.

Using the estimators $\hat{P}$ and $\hat{R}$ with their uncertainties $\sigma\hat{P}$ and $\sigma\hat{R}$ and starting with an initial Q-function Q^0 and corresponding uncertainty σQ^0, e.g., $Q^0 := 0$ and $\sigma Q^0 := 0$, through the update equations (1) and (2) the Q-function and corresponding uncertainty are updated in each iteration and converge to Q^π and σQ^π for policy evaluation and Q^* and σQ^* for policy iteration. Q^* and σQ^* can be used to obtain the function

$$Q_u(s,a) = Q^*(s,a) - \xi \sigma Q^*(s,a), \tag{5}$$

specifying a performance limit which, when the policy π^* is applied to the real MDP, will be exceeded with probability $\Pr(Z(s,a) > Q_u(s,a)) = F(\xi)$, where Z is the (unknown) Q-function of π^* for the real MDP. $F(\xi)$ depends on the distribution class of Q. E.g., if Q is normally distributed, F is the distribution function of the standard normal distribution. Note that a policy based on Q_u, i.e., $\pi_u(s) = \arg\max_a Q_u(s,a)$, does not in general improve the performance limit, as Q_u considers the uncertainty only for one step. In general, Q_u does not represent π_u's Q-function, posing an inconsistency. To use the knowledge of uncertainty for maximizing the performance limit (as opposed to the expectation), the uncertainty needs to be incorporated into the policy-improvement step.

2.2 Uncertainty-Aware Policy Iteration

The policy-improvement step is contained within the Bellman optimality equation as $\max_a Q^m(s,a)$. Alternatively, determining the optimal policy in each iteration as

$$\forall s : \pi^m(s) := \arg\max_a Q^m(s,a) \tag{6}$$

and then updating the Q-function using this policy, i.e.,

$$\forall s,a : Q^m(s,a) := \sum_{s'} \hat{P}(s'|s,a) \left[\hat{R}(s,a,s') + \gamma Q^{m-1}(s', \pi^{m-1}(s)) \right], \tag{7}$$

yields the same solution. To determine a so-called *certain-* or *ξ-optimal* policy that maximizes the performance limit for a given ξ, the update of the policy must not choose the optimal action w.r.t. to the maximum over the Q-values of a particular state but the maximum over the Q-values minus their weighted uncertainty:

$$\forall s : \pi^m(s) := \arg\max_a \left[Q^m(s,a) - \xi\sigma Q^m(s,a)\right]. \tag{8}$$

In each iteration, the uncertainty σQ^m has to be updated as described in sec. 2.1, setting V^m and σV^m as for deterministic policy evaluation.

The parameter ξ controls the influence of the uncertainty on the policy. Choosing a positive ξ yields uncertainty avoiding policies, with increasing ξ a worst-case optimal policy is approached. A negative ξ results in uncertainty seeking behavior.

2.3 Non-Convergence of DUIPI for Deterministic Policy Iteration

While it has been shown that conventional policy iteration in the framework of MDPs is guaranteed to converge to a deterministic policy [9], for ξ-optimal policies derived by the algorithm presented in sec. 2.2 this is not necessarily the case. When considering a Q-value's uncertainty for action selection, there are two effects that contribute to an oscillation of the policy and consequently non-convergence of the corresponding Q-function.

First, there is the effect mentioned in [10] of a bias on $\xi\sigma Q(s,\pi(s))$ being larger than $\xi\sigma Q(s,a), a \neq \pi(s)$, if π is the evaluated policy and $\xi > 0$. DUIPI is not affected by this problem due to the ignorance of covariances between Q and R. Second, there is another effect (by which DUIPI is affected) causing an oscillation when there is a certain constellation of Q-values and corresponding uncertainties of concurring actions. Consider two actions a_1 and a_2 in a state s with similar Q-values but different uncertainties, a_1 having an only slightly higher Q-value but a larger uncertainty. The uncertainty-aware policy improvement step would alter π^m to choose a_2, the action with the smaller uncertainty. However, the fact that this action is inferior might only become obvious in the next iteration when the value function is updated for the altered π^m (and now implying the choice of a_2 in s). In the following policy improvement step the policy will be changed back to choose a_1 in s, since now the Q-function reflects the inferiority of a_2. After the next update of the Q-function, the values for both actions will be similar again, because now the value function implies the choice of a_1 and the bad effect of a_2 affects $Q(s,a_2)$ only once.

2.4 Risk-Reduction by Diversification through Stochastic Policies

With stochastic policies it is possible to construct an update-scheme that is guaranteed to converge, thus solving the problem of non-convergence. Moreover, it is intuitively clear that for $\xi > 0$ ξ-optimal policies should be stochastic as one tries to decrease the risk of obtaining a low long-term reward (because the wrong MDP has been estimated) by diversification.

The resulting algorithm initializes the policy with equiprobable actions. In each iteration, the probability of the best action according to Q_u^m (equation (5)) is increased by $1/m$, m being the current iteration, while the probabilities of all other actions are decreased accordingly:

$$\forall s, a : \pi^m(a|s) := \begin{cases} \min(\pi^{m-1}(a|s) + 1/m, 1), & \text{if } a = a_{Q_u^{m-1}}(s) \\ \frac{\max(1-\pi(s,a_{Q_u^{m-1}}(s))-1/m,0)}{1-\pi(s,a_{Q_u^{m-1}}(s))}\pi^{m-1}(a|s), & \text{otherwise} \end{cases} \tag{9}$$

$a_{Q_u^{m-1}}(s)$ denotes the best action according to Q_u^{m-1}, i.e, $a_{Q_u^{m-1}}(s) = \arg\max_a Q^{m-1}(s,a) - \xi\sigma Q^{m-1}(s,a)$. Due to the harmonically decreasing change rate convergence as well as reachability of all possible policies are ensured.

3 Modeling of Estimators and their Uncertainty

There are several ways of modeling the estimators for the transition probabilities P and the reward R. In the following we will present the frequentist approach using relative frequency as well as a Bayesian approach.

3.1 Frequentist Estimation

In the frequentist paradigm the relative frequency is used as the expected transition probability. The uncertainty of the according multinomial distribution is assumed to be

$$(\sigma\hat{P}(s'|s,a))^2 = \frac{\hat{P}(s'|s,a)(1-\hat{P}(s'|s,a))}{n_{sa}-1}, \tag{10}$$

where n_{sa} denotes the number of observed transitions from (s,a).

Using the same concept for the rewards and assuming a normal distribution, the mean of all observed rewards of a transition (s,a,s') is used as reward expectation, their uncertainties are

$$(\sigma\hat{R}(s,a,s'))^2 = \frac{\mathrm{var}(\hat{R}(s,a,s'))}{n_{sas'}-1}, \tag{11}$$

with $n_{sas'}$ being the number of observed transitions (s,a,s').

Although the estimation of the transition probabilities using relative frequency usually leads to good results in practice, the corresponding uncertainty estimation is problematic if there are only a few observations, because in that case the uncertainties are often underestimated. For instance, if a specific transition is observed twice out of two tries ($n_{sas'} = n_{sa} = 2$), its uncertainty $\sigma\hat{P}(s'|s,a) = 0$.

3.2 Bayesian Estimation

Assuming all transitions from different state-action pairs to be independent of each other and the rewards, the transitions can be modeled as multinomial distributions. In a Bayesian setting, where one assumes a prior distribution over the parameter space $P(s_k|s_i, a_j)$ for given i and j, the Dirichlet distribution with density

$$\Pr(P(s_1|s_i, a_j), \ldots, P(s_{|S|}|s_i, a_j))_{\alpha_{ij1},\ldots,\alpha_{ij|S|}} = \frac{\Gamma(\alpha_{ij})}{\prod_{k=1}^{|S|} \Gamma(\alpha_{ijk})} \prod_{k=1}^{|S|} P(s_k|s_i, a_j)^{\alpha_{ijk}-1}, \tag{12}$$

$\alpha_{ij} = \sum_{k=1}^{|S|} \alpha_{ijk}$, is a conjugate prior with posterior parameters $\alpha^d_{ijk} = \alpha_{ijk} + n_{s_i a_j s_k}$, $\alpha^d_{ij} = \sum_{k=1}^{|S|} \alpha^d_{ijk}$. Choosing the expectation of the posterior distribution as the estimator, i.e., $\hat{P}(s_k|s_i, a_j) = \alpha^d_{ijk}/\alpha^d_{ij}$, the uncertainty of $\hat{P}$ is

$$(\sigma \hat{P}(s_k|s_i, s_j))^2 = \frac{\alpha^d_{ijk}(\alpha^d_{i,j} - \alpha^d_{ijk})}{(\alpha^d_{ij})^2(\alpha^d_{ij} + 1)}. \tag{13}$$

Note that $\alpha_i = 0$ results in a prior that leads to the same estimates and slightly lower uncertainties compared to the frequentist modeling of sec. 3.1. On the other hand, setting $\alpha_i = 1$ leads to a flat, maximum entropy prior that assumes all transitions from a state to all other states equally probable.

Both settings, $\alpha_i = 0$ and $\alpha_i = 1$, represent extremes that we believe are unreasonable for most applications. Instead, we model our prior belief by setting $\alpha_i = \frac{m}{|S|}$, where m is the average number of expected successor states of all state-action pairs and $|S|$ is the total number of states. This choice of α_i realizes an approximation of a maximum entropy prior over a subset of the state space with a size of m states. This way most of the probability is "distributed" among any subset of m states that have actually been observed, the probability of all other (not observed) successor states becomes very low. Compared to the maximum entropy prior with $\alpha_i = 1$ one needs only a few observations for the actually observed successor states to be much more probable than not observed ones. At the same time, the estimation of the uncertainty is not as extreme as the frequentist one, since having made the same observation twice does not cause the uncertainty to become zero. Estimating m from the observations can easily be added.

4 Experiments

We conducted experiments with DUIPI as presented here and the full-matrix algorithm using the frequentist as well as the Bayesian estimators described in the previous section.[1]

4.1 Benchmark: Wet-Chicken 2D

The benchmark problem used was Wet-Chicken 2D, a two-dimensional version of the original Wet-Chicken benchmark [13]. In the original setting a canoeist

[1] Source code for the benchmark problem as well as a DUIPI implementation is available at `http://ahans.de/publications/icann2009/`.

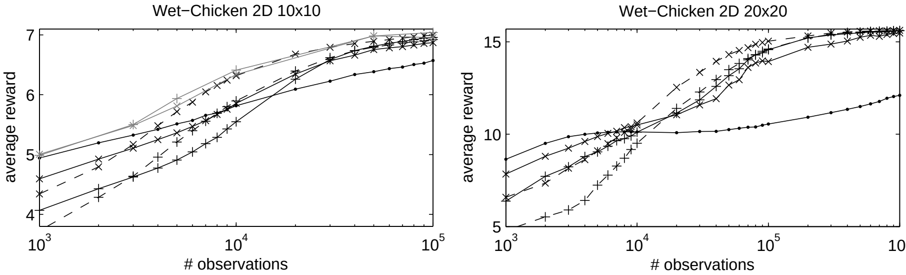

Fig. 1. Performance of policies generated using standard policy iteration ('•' marks), DUIPI (black lines), and the full-matrix method [10] (gray lines). $\xi = 0.5$ is indicated by '×' marks, $\xi = 1$ by '+' marks. Solid lines represent policies generated using frequentist estimators, dashed lines represent policies generated using Bayesian estimation.

paddles on a one-dimensional river with length l and flow velocity $v = 1$. At position $x = l$ of the river there is a waterfall. Starting at position $x = 0$ the canoeist has to try to get as near as possible to the waterfall without falling down. If he falls down, he has to restart at position $x = 0$. The reward increases linearly with the proximity to the waterfall and is given by $r = x$. The canoeist has the possibility to drift $(x - 0 + v = x + 1)$, to hold the position $(x - 1 + v = x)$, or to paddle back $(x - 2 + v = x - 1)$. River turbulence of size $s = 2.5$ causes the state transitions to be stochastic. Thus, after having applied the canoeist's action to his position (also considering the flow of the river), the new position is finally given by $x' = x + n$, where $n \in [-s, s]$ is a uniformly distributed random value. For the two-dimensional version the river is extended by a width w. Accordingly, there are two additional actions available to the canoeist, one to move the canoe to the left and one to move it to the right by one unit. The position of the canoeist is now denoted by (x, y), the (re-)starting position is $(0, 0)$. The velocity of the flow v and the amount of turbulence s depend on y: $v = 3y/w$ and $s = 3.5 - v$. In the discrete problem setting, which we use here, x and y are always rounded to the next integer value. While on the left edge of the river the flow velocity is zero, the amount of turbulence is maximal; on the right edge there is no turbulence (in the discrete setting), but the velocity is too high to paddle back.

4.2 Results

We performed experiments with a river size of 10x10 (100 states) and 20x20 (400 states). For both settings a fixed number of observations was generated using random exploration. The observations were used as input to generate policies using the different algorithms. The discount factor was chosen as $\gamma = 0.95$. Each resulting policy was evaluated over 100 episodes with 1000 steps each. The results are summarized in fig. 1 (averaged over 100 trials). For clarity only the results of stochastic policies are shown (except for $\xi = 0$, i.e., standard policy iteration), they performed better than the deterministic ones in all experiments. Usually a method like DUIPI aims at quantile optimization, i.e., reducing the

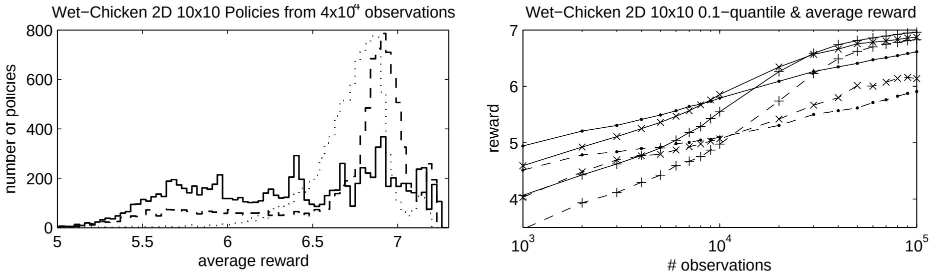

Fig. 2. Left: histograms of average rewards of 10^4 policies with $\xi = 0$ (solid), $\xi = 1$ (dashed), and $\xi = 2$ (dotted). For the generation of each policy 4×10^4 observations were used. Right: mean (solid) and 0.1-quantile (dashed) average rewards of policies with $\xi = 0$ ('$\bullet$' marks), $\xi = 0.5$ ('$\times$' marks), and $\xi = 1$ ('$+$' marks).

probability of generating very poor policies at the expense of a lower expected average reward. However, in some cases it is even possible to increase the expected performance, when the MDP exhibits states that are rarely visited but potentially result in a high reward. For Wet-Chicken states near the waterfall have those characteristics. An uncertainty unaware policy would try to reach those states if there are observations leading to the conclusion that the probability of falling down is low, which in fact is high. In [10] this is reported as "border-phenomenon", which by our more general explanation is included. Due to this effect it is possible to increase the average performance using uncertainty aware methods for policy generation, which can be seen from the figure. For small numbers of observations and high ξ-values DUIPI performs worse as in those situations the action selection in the iteration is dominated by the uncertainty of the Q-values and not the Q-values themselves. This leads to a preference of actions with low uncertainty, the Q-values play only a minor role. This effect is increased by the fact that due to random exploration most observations are near the beginning of the river, where the immediate reward is low. Using a more intelligent exploration scheme could help to overcome this problem. Due to the large computational and memory requirements the full-matrix method could not be applied to the problem with river size 20x20. Moreover, results of the full-matrix version with Bayesian estimation are not shown as they would not have been distinguishable in the figure.

Fig. 2 compares uncertainty aware and unaware methods. Considering the uncertainty reduces the amount of poor policies and even increases the expected performance ($\xi = 0.5$). Setting $\xi = 1$ results in an even lower probability for poor policies at the expense of a lower expected average reward.

Table 1. Computation times to generate a policy using a single core of an Intel Core 2 Quad Q9550 processor.

method	Wet-Chicken 5x5	Wet-Chicken 10x10	Wet-Chicken 20x20
full-matrix	5.61 s	1.1×10^3 s	—
DUIPI	0.002 s	0.034 s	1.61 s

5 Conclusion

In this paper, we presented DUIPI, a computationally very feasible algorithm for incorporation of uncertainty into the Bellman iteration. It only considers the diagonal of the covariance matrix encoding the covariance. While this causes the algorithm to be only approximate, it also decreases its complexity, decreasing the computational requirements by orders of magnitude. Moreover, we proposed a Bayesian parameter estimation that incorporates prior knowledge about the number of successor states. Our experiments show that DUIPI increases the robustness and performance of policies generated for MDPs whose exact parameters are unknown and estimated from only a fixed set of observations. In industrial applications observations are often expensive and arbitrary exploration not possible, we therefore believe that for those applications knowledge of uncertainty is crucial. Future work will consider application of UP to RL algorithms involving function approximation and utilizing knowledge of uncertainty for efficient exploration.

References

1. G. Calafiore and L. El Ghaoui. On distributionally robust chance-constrained linear programs. In *Optimization Theory and Applications*, 2006.
2. G. D'Agostini. *Bayesian Reasoning in Data Analysis: A Critical Introduction.* World Scientific Publishing, 2003.
3. E. Delage and S. Mannor. Percentile optimization in uncertain Markov decision processes with application to efficient exploration. In *Proc. of the Int. Conf. on Machine Learning*, 2007.
4. Y. Engel, S. Mannor, and R. Meir. Bayes meets Bellman: the Gaussian process approach to temporal difference learning. In *Proc. of the Int. Conf. on Machine Learning*, 2003.
5. Y. Engel, S. Mannor, and R. Meir. Reinforcement learning with Gaussian processes. In *Proc. of the Int. Conf. on Machine Learning*, 2005.
6. M. Ghavamzadeh and Y. Engel. Bayesian policy gradient algorithms. In *Advances in Neural Information Processing Systems*, 2006.
7. M. Ghavamzadeh and Y. Engel. Bayesian actor-critic algorithms. In *Proc. of the Int. Conf. on Machine Learning*, 2007.
8. A. Nilim and L. El Ghaoui. Robustness in Markov decision problems with uncertain transition matrices. In *Advances in Neural Information Processing Systems*, 2003.
9. M.L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* John Wiley & Sons Canada, Ltd., 1994.
10. D. Schneegass, S. Udluft, and T. Martinetz. Uncertainty propagation for quality assurance in reinforcement learning. In *Proc. of the Int. Joint Conf. on Neural Networks*, 2008.
11. A.L. Strehl and M.L. Littman. An empirical evaluation of interval estimation for markov decision processes. In *16th IEEE Int. Conf. on Tools with Artificial Intelligence*, pages 128–135, 2004.
12. R.S. Sutton and A.G. Barto. *Reinforcement Learning: An Introduction.* MIT Press, 1998.
13. V. Tresp. The wet game of chicken. *Siemens AG, CT IC 4, Technical Report*, 1994.