Bayes Adaptive Reinforcement Learning versus Off-line Prior-based Policy Search: an Empirical Comparison

Michaël Castronovo M.CASTRONOVO@ULG.AC.BE
University of Liège, Institut Montefiore, B28, B-4000 Liège, BELGIUM

Damien Ernst DERNST@ULG.AC.BE
University of Liège, Institut Montefiore, B28, B-4000 Liège, BELGIUM

Raphaël Fonteneau RAPHAEL.FONTENEAU@ULG.AC.BE
University of Liège, Institut Montefiore, B28, B-4000 Liège, BELGIUM

Abstract

This paper addresses the problem of decision making in unknown finite Markov Decision Processes (MDPs). The uncertainty about the MDPs is modelled, using a prior distribution over a set of candidate MDPs. The performance criterion is the expected sum of discounted rewards collected over an infinite length trajectory. Time constraints are defined as follows: (i) an off-line phase with a given time budget, which can be used to exploit the prior distribution and (ii) at each time step of the on-line phase, decisions have to be computed within a given time budget. In this setting, two decision-making strategies are compared. Firstly, OPPS, which is a recently proposed meta-learning scheme that mainly exploits the off-line phase to perform the policy search, as well as BAMCP—that is a state-of-the-art model-based Bayesian reinforcement learning algorithm, which mainly exploits the on-line time budget. These approaches are empirically compared in a real Bayesian setting, with their performances computed over a large set of problems. As far as this particular area of study is concerned, it is the first time that this is done in the Reinforcement Learning literature. Several settings are considered by varying the prior distribution and the distribution from which test problems are drawn. The main finding of these experiments is that there may be a significant benefit of having an off-line prior-based optimization phase, in the case of informative and accurate priors, especially when on-line time constraints are tight.

Appearing in *Proceedings of BENELEARN 2014*.

1. Introduction

Optimally interacting with an unknown Markov Decision Process (MDP) remains a challenging Reinforcement Learning (RL) problem (Buşoniu et al., 2010). At the heart of this challenge lies the so-called Exploration/Exploitation (E/E) dilemma: on one hand, the agent needs to collect relevant data by exploring the environment, at the cost of taking bad decisions in the short term, while exploiting its current knowledge, facing the risk to take sub-optimal actions in the long term.

In the last fifteen years, Bayesian RL (Dearden et al., 1999; Strens, 2000) developed an interesting method to deal with the fact that the actual MDP is unknown. It assumes a prior distribution over a set of candidate MDPs from which the actual MDP is likely to be drawn. When interacting with the actual MDP, a posterior distribution is maintained, given the prior and the transitions observed so far. Such a posterior is used at each time-step to compute near-optimal Bayesian decisions, as a strategy to deal with the E/E dilemma. Model-based Bayesian RL methods maintain a posterior distribution over transition models (Ross & Pineau, 2008; Poupart, 2008; Asmuth et al., 2009; Hennig et al., 2009; Fard & Pineau, 2010; Ross et al., 2011). On the other hand, the model-free Bayesian RL methods do not explicitly maintain a posterior over transition models, but rather value functions from which a decision can be extracted (see

e.g. (Dearden et al., 1998; Engel et al., 2003; Engel et al., 2005a; Engel et al., 2005b; Ghavamzadeh & Engel, 2006; Ghavamzadeh & Engel, 2007).

Recently, Guez et al. (Guez et al., 2012) have introduced the BAMCP algorithm (for Bayes-adaptive Monte Carlo planning), a model-based Bayesian RL approach, which combines the principle of the UCT—Upper Confidence Trees—with sparse sampling methods, and obtained state-of-the-art performances. At the same time, Castronovo et al. (Castronovo et al., 2012) proposed an algorithm that exploits a prior distribution, in an off-line phase, by solving a policy search problem in a wide space of candidate, index-based E/E strategies, and by applying the obtained strategy on the actual MDP afterwards. The purpose of this paper is to empirically compare those two approaches in a "real Bayesian setting". In order to achieve this, several MDP distributions are considered, which can either be used as a prior distribution or as a test distribution, from which test problems are drawn. Several possible configurations in terms of prior/test distribution association are also considered, in order to observe the effect of the "flatness" of the prior distributions or their "accuracy" on the performances of the algorithms. Moreover, in order to be objective, comparisons will take into account the minimal computation time required to run each of these algorithms. Our experiments mainly show that exploiting a prior distribution in an off-line phase makes sense in the context of informative and accurate priors, especially for problems where on-line time constraints are tight.

The paper is organized in the following manner: Section 2 formalizes the problem addressed in this paper. Section 3 presents the experimental protocol and the empirical results. Section 4 discusses the obtained results, and finally Section 5 concludes the paper.

2. Problem Statement

The goal of this paper is to compare two Reinforcement Learning (RL) strategies, in the presence of a prior distribution. First we describe the RL setting in Section 2.1. Then the prior distribution assumption is formalized in Section 2.2, and the basics of the BAMCP and OPPS approaches are briefly described. Section 2.3 formalizes the computational time constraints that these algorithms must satisfy, and Section 2.4 explains the specificities of our empirical evaluation.

2.1. Reinforcement Learning

Let $M = (X, U, f(\cdot), \rho_M, \rho_{M,0}(\cdot), \gamma)$ be a given unknown MDP, where $X = \{x^{(1)}, \ldots, x^{(n_X)}\}$ denotes its finite state space and $U = \{u^{(1)}, \ldots, u^{(n_U)}\}$ its finite action space. When the MDP is in state x_t at time t and action u_t is selected, the agent moves instantaneously to a next state x_{t+1} with a probability of $P(x_{t+1}|x_t, u_t) = f(x_t, u_t, x_{t+1})$. An instantaneous deterministic, bounded reward $r_t = \rho(x_t, u_t, x_{t+1}) \in [R_{\min}, R_{\max}]$ is observed simultaneously. In this paper, the reward function ρ is assumed to be fully known, which is often true in practice.

Let $H_t = (x_0, u_0, r_0, x_1, \cdots, x_{t-1}, u_{t-1}, r_{t-1}, x_t)$ denote the history observed until time t. An E/E strategy is a stochastic policy h that, given the current state x_t, returns an action $u_t \sim h(H_t)$. Given a probability distribution over initial states $\rho_{M,0}(\cdot)$, the expected return of a given E/E strategy h with respect to the MDP M can be defined as follows:

$$J_M^h = \mathop{\mathbb{E}}_{x_0 \sim \rho_{M,0}(\cdot)} [\mathcal{R}_M^h(x_0)],$$

where $\mathcal{R}_M^h(x_0)$ is the stochastic discounted sum of rewards received when applying the E/E strategy h, starting from an initial state x_0, defined as indicated below:

$$\mathcal{R}_M^h(x_0) = \sum_{t=0}^{+\infty} \gamma^t r_t,$$

where the discount factor γ belongs to $[0, 1)$. Within this setting, reinforcement learning amounts in finding a policy h^* which leads to the maximization of J_M^h:

$$h^* \in \arg\max_h \quad J_M^h \ .$$

2.2. Prior Distribution over a Set of Candidate Models

In the context where the actual MDP is initially unknown, the Bayesian RL techniques propose to model the uncertainty about the actual model, using a probability distribution. This amounts to assuming that the actual MDP to be played is drawn from a distribution $p_{\mathcal{M}}^0(\cdot)$. In the so-called model-based Bayesian RL setting, this prior distribution is assumed to be known. In this paper, it is assumed that there is access to a prior distribution $p_{\mathcal{M}}^0(\cdot)$ over a set of MDPs $\mathcal{M}$. It is further assumed that:

- One can easily draw MDPs models from $p_{\mathcal{M}}^0(\cdot)$;
- One can easily compute the posterior distribution from $p_{\mathcal{M}}^0(\cdot)$ given the observation of an history H_t and the prior distribution.

Using these assumptions, the goal is to determine an E/E strategy h which leads to the maximization of the expected return over the set of transition models $\mathcal{M}$:

$$h^* \in \arg\max_h \mathop{\mathbb{E}}_{M\sim p^0_{\mathcal{M}}(\cdot)} \left[J^h_M\right] \ .$$

In this paper, two algorithms that can take advantage of such a prior are compared; these are the BAMCP and OPPS algorithms.

2.2.1. The BAMCP Algorithm

The BAMCP (Bayes-adaptive Monte Carlo planning) algorithm is a state-of-the-art performance Bayesian RL algorithm, originally proposed in (Guez et al., 2012). The principle of this algorithm is to adapt the UCT (Upper Confidence bounds applied to Trees, see (Kocsis & Szepesvári, 2006)) principle for planning in a Bayes-adaptive MDP, also called the belief-augmented MDP, which is a MDP obtained when considering augmented states made of the concatenation of the actual state and the posterior. The BAMCP algorithm is made computationally tractable by using a sparse sampling strategy, which avoids sampling a model from the posterior distribution at every node of the planification tree. In practice, given a prior $p^0_{\mathcal{M}}(.)$ and a history H_t, the BAMCP algorithm computes a policy h^{BAMCP}_K based on the building of a planification tree with exactly K nodes, from which a decision is outputted:

$$u_t \sim h^{BAMCP}_K \left(H_t, p^0_{\mathcal{M}}(.)\right) \ .$$

Note that, as the number of node expansions K increases to infinity, the decision computed by the BAMCP algorithm converges towards Bayesian optimality.

2.2.2. The OPPS Algorithm

The Off-line, Prior-based Policy Search (OPPS) algorithm was originally introduced in (Castronovo et al., 2012). The OPPS approach works as follows: (i) a set of candidate E/E strategies $\mathcal{S}$ is built, and (ii) a policy search scheme is run over the set of strategies. The strategy space is obtained by considering index-based strategies, where the index is generated using small formulas, combining the standard mathematical operators with standard RL features (i.e., value functions). The search of an optimal E/E strategy is formalized as a multi-armed bandit problem, with a number of arms being equal to the number of candidate E/E strategies. Pulling an arm amounts to draw a MDP from the prior, and to proceed with one single run of the candidate E/E corresponding to that arm. Formally, the OPPS algorithm computes — during the off-line phase — a policy $h^{OPPS}_{\mathcal{S}}$ from which decisions are extracted on-line, given the prior $p^0_{\mathcal{M}}(.)$ and the history H_t:

$$u_t \sim h^{OPPS}_{\mathcal{S}}(H_t, p^0_{\mathcal{M}}(.))$$

where

$$h^{OPPS}_{\mathcal{S}} \in \arg\max_{s\in\mathcal{S}} \mathop{\mathbb{E}}_{M\sim p^0_{\mathcal{M}}(\cdot)} \left[J^s_M\right] \ .$$

In this paper, the set of variables from which formulas are built is slightly different than the one used in (Castronovo et al., 2012). Such a set is fully described in Appendix A.

2.3. Time Constraints

Bayesian RL has acquired the reputation of being computationally intensive, mainly because of the incorporation of the posterior updates in the planification phase. In this paper, we propose to explicitly formalize the computational time budget allocated at every phase of the use of the algorithms. Thus, two types of time constraints are considered:

- an "off-line" time period B_{-1}, corresponding to a phase when the prior distribution is available to the agent, but the actual MDP is not yet available for interaction;
- a sequence of "on-line" time periods is considered $B_0, B_1 \ldots$, where, for all $t \in \mathbb{N}$, B_t corresponds to the time period available to compute a decision $u_t \in U$ given the prior $p^0_{\mathcal{M}}(.)$ and the history H_t observed so far.

2.4. Bayesian Empirical Evaluation

In this paper, we propose a real Bayesian empirical evaluation, in the sense that we compare the algorithms on a large set of problems drawn according to a test probability distribution. Such a distribution can be similar ("accurate") or different ("inaccurate") from the prior. Formally, for each experiment, a prior distribution is considered $p^0_{\mathcal{M}}(\cdot)$, which is given to the algorithms as an input, and a test distribution $p_{\mathcal{M}}(\cdot)$, which is used to draw test problems, on which each algorithm is evaluated. As far as this area of study is concerned, this is the first time that the Bayesian RL algorithms are compared on average over a large set of problems, rather than on standard benchmarks.

3. Experiments

Each experiment is characterized by the following:

- A prior distribution $p^0_{\mathcal{M}}(\cdot)$,
- A test distribution $p_{\mathcal{M}}(\cdot)$,
- An off-line time budget B_{-1},
- On-line time budgets $B_0, B_1, \cdots$ for computing decisions applied on the actual MDP.

The goal of those experiments is to identify the influence of the above mentioned elements on the performance of the algorithms, and, consequently, to identify the domain of excellence of each algorithm.

Subsection 3.1 describes the experimental protocol used to compare the algorithms described in Section 2.2. Subsection 3.2 defines accurately the MDP distributions considered in the experiments presented in Subsection 3.3.

3.1. The Experimental Protocol

For each algorithm:

- a pool of $10,000$ MDPs is drawn from $p_{\mathcal{M}}(.)$;
- one single run of the algorithm is simulated on each MDP of the pool;
- its empirical expected average of discounted returns is computed.

Trajectories are truncated after T steps, where T is defined as follows:

$$T = \left\lceil \frac{\frac{\epsilon \times (1-\gamma)}{R_{max}}}{\log \gamma} \right\rceil \text{ with } \epsilon = 0.001.$$

The mean μ is measured and the standard deviation σ of the set of observed returns. This data allows us to compute the 95% confidence interval of $J^{h(p^0_{\mathcal{M}}(\cdot))}_{p_{\mathcal{M}}(\cdot)}$:

$$J^{h(p^0_{\mathcal{M}}(\cdot))}_{p_{\mathcal{M}}(\cdot)} \in \left[\mu - \frac{2\sigma}{\sqrt{10,000}}; \mu + \frac{2\sigma}{\sqrt{10,000}}\right] \text{ with probability at least } 95\%.$$

Since each MDP distribution described below can be used, either as a prior distribution $p^0_{\mathcal{M}}(\cdot)$ or as a test distribution $p_{\mathcal{M}}(\cdot)$, the process is repeated for each possible combination.

3.2. MDP Distributions

The MDP distributions introduced in this paper are inspired from the well-known five-state chain MDP (Strens, 2000). For all the MDP distributions considered in this paper, the set of candidate MDPs shares the same state space X, action space U, reward function ρ_M, initial state distribution $\rho_{M,0}(\cdot)$ and discount factor γ. In our experiments, $X = \{1, 2, 3, 4, 5\}$, $U = \{1, 2, 3\}$, $\gamma = 0.95$, $x_0 = 1$ with probability 1 and the reward function ρ_M is defined as follows:

$$\begin{aligned} \forall (x,u) \in X \times U, \rho_M(x,u,1) &= 2.0 \\ \forall (x,u) \in X \times U, \rho_M(x,u,5) &= 10.0 \\ \forall (x,u) \in X \times U, y \in \{2,3,4\}, \rho_M(x,u,y) &= 0.0. \end{aligned}$$

In this context, a MDP is entirely specified by its transition matrix. Therefore, the probability distribution over sets of candidate transition matrices is defined, using the Flat Dirichlet Multinomial (FDM) distributions, which are widely used in the Bayesian RL, mostly because their Bayes update is straightforward. One independent Dirichlet distribution per state-action pair (x, u) is assumed, which leads to a density d_{FDM}:

$$d_{FDM}(\mu; \boldsymbol{\theta}) = \prod_{x,u} D(\mu_{x,u}; \theta_{x,u})$$

where $D(\cdot;\cdot)$ are independent Dirichlet distributions. The parameter $\boldsymbol{\theta}$ gathers all the counters of observed transitions $\theta^t_{x,u}$ until time t, including $\theta^0_{x,u}$ which represents a priori observations.

The density of $p_{\mathcal{M}}(\cdot)$ is therefore defined as:

$$d_{p_{\mathcal{M}}(\cdot)}(\mu, \boldsymbol{\theta}) = d_{FDM}(\mu; \boldsymbol{\theta})$$

Consequently, a MDP distribution is parameterised by $\boldsymbol{\theta}$, and will be denoted by $p^{\boldsymbol{\theta}}(\cdot)$. In the following section, we introduce four MDP distributions, the "Generalized Chain" distribution, the "Optimistic Generalized Chain" distribution, the "Pessimistic Generalized Chain" distribution and the "Uniform" distribution.

3.2.1. Generalized Chain Distribution

This MDP distribution is a generalisation of the well-known Chain MDP. For each action, two different outcomes are possible:

- The agent moves from state x to state $x + 1$ (or remains in state x when $x = 5$) or;
- The agent "slips" and goes back to the initial state.

The probabilities associated with those outcomes are drawn uniformly. Formally, the $\boldsymbol{\theta}^{GC}$ parameter characterising the corresponding $p^{\boldsymbol{\theta}^{GC}}(\cdot)$ distribution is defined as follows:

$$\forall u \in U : \theta^{GC}_{1,u} = [1, 1, 0, 0, 0]$$
$$\forall u \in U : \theta^{GC}_{2,u} = [1, 0, 1, 0, 0]$$
$$\forall u \in U : \theta^{GC}_{3,u} = [1, 0, 0, 1, 0]$$
$$\forall u \in U : \theta^{GC}_{4,u} = [1, 0, 0, 0, 1]$$
$$\forall u \in U : \theta^{GC}_{5,u} = [1, 0, 0, 0, 1]$$

3.2.2. Optimistic Generalized Chain Distribution

This distribution is an alternative to the Generalized Chain MDPs, where higher weights are put on transitions, allowing the agent to move forward in the chain. Formally, the $\boldsymbol{\theta}^{OGC}$ parameter characterising the corresponding $p^{\boldsymbol{\theta}^{OGC}}(\cdot)$ distribution is defined as follows:

$$\forall u \in U : \theta^{OGC}_{1,u} = [1, 5, 0, 0, 0]$$
$$\forall u \in U : \theta^{OGC}_{2,u} = [1, 0, 5, 0, 0]$$
$$\forall u \in U : \theta^{OGC}_{3,u} = [1, 0, 0, 5, 0]$$
$$\forall u \in U : \theta^{OGC}_{4,u} = [1, 0, 0, 0, 5]$$
$$\forall u \in U : \theta^{OGC}_{5,u} = [1, 0, 0, 0, 5]$$

3.2.3. Pessimistic Generalized Chain Distribution

This distribution is an alternative to the Generalized Chain MDPs, where higher weights are put on transitions, moving the agent to the initial state. Formally, the $\boldsymbol{\theta}^{PGC}$ parameter characterising the corresponding $p^{\boldsymbol{\theta}^{PGC}}(\cdot)$ distribution is defined as follows:

$$\forall u \in U : \theta^{PGC}_{1,u} = [5, 1, 0, 0, 0]$$
$$\forall u \in U : \theta^{PGC}_{2,u} = [5, 0, 1, 0, 0]$$
$$\forall u \in U : \theta^{PGC}_{3,u} = [5, 0, 0, 1, 0]$$
$$\forall u \in U : \theta^{PGC}_{4,u} = [5, 0, 0, 0, 1]$$
$$\forall u \in U : \theta^{PGC}_{5,u} = [5, 0, 0, 0, 1]$$

3.2.4. Uniform Distribution

All transition probabilities are drawn uniformly. Formally, the $\boldsymbol{\theta}^{U}$ parameter characterising the corresponding $p^{\boldsymbol{\theta}^{U}}(\cdot)$ distribution is defined as follows:

$$\forall x \in X, u \in U : \theta^{U}_{x,u} = [1, 1, 1, 1, 1]$$

Finally, note that unlike the original chain MDP, in which action 1 is optimal in any given state, the optimal behaviour in any MDP drawn according to one of these distributions is not defined a priori, as it changes from one MDP to another.

3.3. The Results of the Experiments

Several experiments are presented, where different prior distribution / test distribution combinations are considered.

Concerning OPPS, four different strategy spaces are considered. The set of variables, operators and constants has been fixed once and for all. The four strategy spaces differ only by the maximal length of the small formulas, which can be built from them. Those spaces were named $\mathbb{F}_n$, where n is the maximal length of the formulas of the corresponding strategy space. The implementation of OPPS used in these experiments differs from the one of (Castronovo et al., 2012) by the chosen set of variables. These variables are described in the Appendix A.

Concerning BAMCP, the default parameters provided by Guez et al. in (Guez et al., 2012) were used. Several instances of BAMCP are built by varying the number of nodes, which are created at each time-step. This parameter has been denoted by K.

Our experiments are organized in four different parts, one for each possible test distribution, i.e. the distribution from which test problems are drawn. In each part, we present a table of experimental results, obtained when the prior and test distributions are identical, comparing the algorithms, in term of performances and minimal required off-line / on-line time budgets. In addition, a figure is joined, comparing the performances of the approaches for different prior distributions.

3.3.1. "Generalized Chain" Test Distribution

Agent	Offline time	Online time	Mean score
OPPS ($\mathbb{F}_3$)	$\sim$ 6h	$\sim$ 40ms	42.29 ± 0.45
OPPS ($\mathbb{F}_4$)	$\sim$ 6h	$\sim$ 42ms	41.89 ± 0.41
OPPS ($\mathbb{F}_5$)	$\sim$ 6h	$\sim$ 42ms	41.89 ± 0.41
BAMCP ($K = 1$)	$\sim$ 1ms	$\sim$ 7ms	31.71 ± 0.23
BAMCP ($K = 10$)	$\sim$ 1ms	$\sim$ 54ms	33.23 ± 0.26
BAMCP ($K = 25$)	$\sim$ 1ms	$\sim$ 136ms	33.26 ± 0.26
BAMCP ($K = 50$)	$\sim$ 1ms	$\sim$ 273ms	33.73 ± 0.26
BAMCP ($K = 100$)	$\sim$ 1ms	$\sim$ 549ms	33.99 ± 0.27
BAMCP ($K = 250$)	$\sim$ 1ms	$\sim$ 2s	34.02 ± 0.26
BAMCP ($K = 500$)	$\sim$ 1ms	$\sim$ 3s	34.27 ± 0.26

Table 1. Comparison with prior "Generalized Chain" on "Generalized Chain"

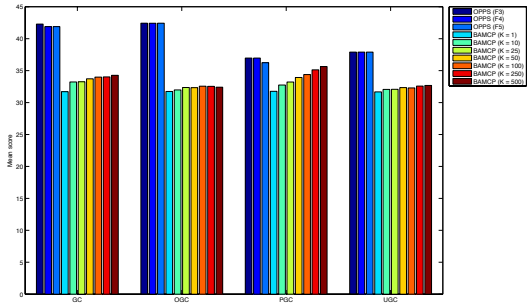

Figure 1. Comparison on "Generalized Chain" distribution

Table 1 shows that OPPS outperforms BAMCP in every single case, even for higher on-line time budgets. The choice of the prior has a significant impact on the performances of OPPS, as shown by Figure 1. The "Generalized Chain" and "Optimistic Generalized Chain" priors show similar performances for OPPS, while the "Uniform Generalized Chain" prior degrades them. On its side, BAMCP is steady except for the "Pessimistic Generalized Chain" prior, which has a positive effect on its performances, contrary to OPPS.

3.3.2. "Optimistic Generalized Chain" Test Distribution

Agent	Offline time	Online time	Mean score
OPPS ($\mathbb{F}_3$)	$\sim$ 6h	$\sim$ 44ms	110.48 ± 0.61
OPPS ($\mathbb{F}_4$)	$\sim$ 6h	$\sim$ 44ms	110.51 ± 0.61
OPPS ($\mathbb{F}_5$)	$\sim$ 6h	$\sim$ 45ms	110.48 ± 0.61
BAMCP ($K = 1$)	$\sim$ 1ms	$\sim$ 7ms	92.71 ± 0.58
BAMCP ($K = 10$)	$\sim$ 1ms	$\sim$ 56ms	93.97 ± 0.57
BAMCP ($K = 25$)	$\sim$ 1ms	$\sim$ 138ms	94.24 ± 0.58
BAMCP ($K = 50$)	$\sim$ 1ms	$\sim$ 284ms	94.31 ± 0.57
BAMCP ($K = 100$)	$\sim$ 1ms	$\sim$ 555ms	94.59 ± 0.57
BAMCP ($K = 250$)	$\sim$ 1ms	$\sim$ 2s	95.06 ± 0.57
BAMCP ($K = 500$)	$\sim$ 1ms	$\sim$ 3s	95.27 ± 0.58

Table 2. Comparison with prior "Optimistic Generalized Chain" on "Optimistic Generalized Chain"

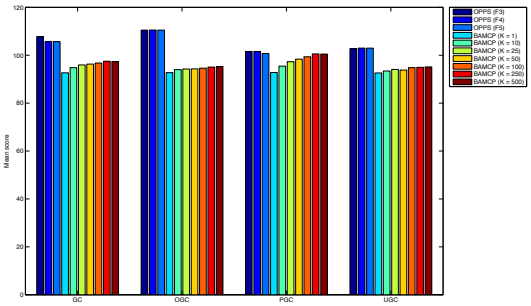

Figure 2. Comparison on "Optimistic Generalized Chain" distribution

Table 2 shows that OPPS clearly outperforms BAMCP, even for pretty high time budgets. However, in Figure 2, we can see that BAMCP becomes more competitive when using the "Pessimistic Generalized Chain" prior distribution. In this case, BAMCP is near OPPS performances.

3.3.3. "Pessimistic Generalized Chain" Test Distribution

Agent	Offline time	Online time	Mean score
OPPS ($\mathbb{F}_3$)	$\sim$ 5h	$\sim$ 37ms	35.89 ± 0.06
OPPS ($\mathbb{F}_4$)	$\sim$ 5h	$\sim$ 39ms	35.89 ± 0.06
OPPS ($\mathbb{F}_5$)	$\sim$ 5h	$\sim$ 38ms	35.83 ± 0.06
BAMCP ($K = 1$)	$\sim$ 1ms	$\sim$ 6ms	33.77 ± 0.07
BAMCP ($K = 10$)	$\sim$ 1ms	$\sim$ 54ms	33.97 ± 0.06
BAMCP ($K = 25$)	$\sim$ 1ms	$\sim$ 133ms	34.1 ± 0.06
BAMCP ($K = 50$)	$\sim$ 1ms	$\sim$ 265ms	34.21 ± 0.06
BAMCP ($K = 100$)	$\sim$ 1ms	$\sim$ 536ms	34.37 ± 0.06
BAMCP ($K = 250$)	$\sim$ 1ms	$\sim$ 2s	34.62 ± 0.06
BAMCP ($K = 500$)	$\sim$ 1ms	$\sim$ 3s	34.9 ± 0.06

Table 3. Comparison with prior "Pessimistic Generalized Chain" on "Pessimistic Generalized Chain"

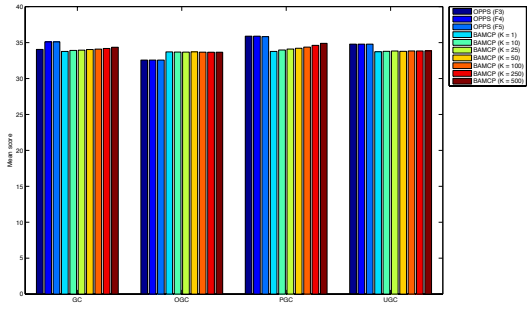

Figure 3. Comparison on "Pessimistic Generalized Chain" distribution

Table 3 shows that OPPS and BAMCP share similar performances, even if BAMCP stays behind. Nevertheless, BAMCP requires on-line time budgets that are eighty times higher than the one required by OPPS, in order to get a slightly lower score. As shown in Figure 3, this difference remains in all cases except for the "Optimistic Generalized Chain" case where BAMCP clearly outperforms OPPS.

3.3.4. "Uniform Generalized Chain" Test Distribution

Agent	Offline time	Online time	Mean score
OPPS ($\mathbb{F}_3$)	$\sim$ 8h	$\sim$ 52ms	57.37 ± 0.38
OPPS ($\mathbb{F}_4$)	$\sim$ 8h	$\sim$ 53ms	57.37 ± 0.38
OPPS ($\mathbb{F}_5$), UGC)	$\sim$ 8h	$\sim$ 51ms	57.37 ± 0.38
BAMCP ($K = 1$)	$\sim$ 1ms	$\sim$ 6ms	47.92 ± 0.29
BAMCP ($K = 10$)	$\sim$ 1ms	$\sim$ 52ms	48.81 ± 0.3
BAMCP ($K = 25$)	$\sim$ 1ms	$\sim$ 132ms	48.95 ± 0.3
BAMCP ($K = 50$)	$\sim$ 1ms	$\sim$ 256ms	49.3 ± 0.3
BAMCP ($K = 100$)	$\sim$ 1ms	$\sim$ 521ms	49.39 ± 0.31
BAMCP ($K = 250$)	$\sim$ 1ms	$\sim$ 2s	50.08 ± 0.31
BAMCP ($K = 500$)	$\sim$ 1ms	$\sim$ 3s	50.06 ± 0.31

Table 4. Comparison with prior "Uniform" on "Uniform"

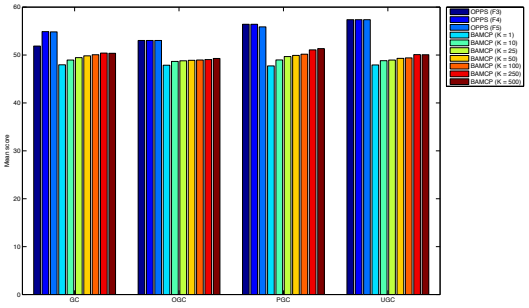

Figure 4. Comparison on "Uniform Generalized Chain" distribution

Both Table 4 and Figure 4 show a clear victory for OPPS for any prior distribution, even with pretty high on-line time budgets. We can also notice that OPPS is more efficient when using the correct prior distribution.

4. Discussion

As a general remark, it is observed that OPPS performs better than BAMCP, even for high on-line time budgets, at the cost of several hours of offline computation time. However, we can notice that BAMCP was a decent challenger in the case of "Pessimistic Generalized Chain" distribution.

Regarding the accuracy of the prior, it appears that using a prior distribution, which differs from the test problem distribution impacts the performances of OPPS in a negative manner, which is expected, since OPPS performs policy search, using the prior. This impact is strengthened in the case of a tight test distribution ("Generalized Chain", "Optimistic Generalized Chain" and "Pessimistic Generalized Chain"). Thanks to the posterior update, the performance of BAMCP seems less affected by a prior inaccuracy.

5. Conclusion

An extensive experimental comparison between two different Bayesian approaches was presented, exploiting either off-line or on-line time budgets, in order to interact efficiently with an unknown MDP. Our experiments suggest that: (i) exploiting a prior distribution in an off-line phase is never a bad idea, even for problems where on-line time constraints are loose, whereas (ii) when on-line time budget are less constrained, maintaining a posterior distribution definitely decreases the impact of an inaccurate prior on the performances of the agent.

Acknowledgments

Michaël Castronovo acknowledges the financial support of the FRIA and the CECI[1]. Raphael Fonteneau is a postdoctoral fellow of the FRS-FNRS.

References

Asmuth, J., Li, L., Littman, M., Nouri, A., & Wingate, D. (2009). A Bayesian sampling approach to exploration in reinforcement learning. *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence (UAI)* (pp. 19–26). AUAI Press.

Buşoniu, L., Babuška, R., De Schutter, B., & Ernst, D. (2010). *Reinforcement learning and dynamic programming using function approximators*. Boca Raton, Florida: CRC Press.

Castronovo, M., Maes, F., Fonteneau, R., & Ernst, D. (2012). Learning exploration/exploitation strategies for single trajectory reinforcement learning. *Journal of Machine Learning Research (JMLR)*, *24*, 1–9.

Dearden, R., Friedman, N., & Andre, D. (1999). Model based Bayesian exploration. *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence (UAI)* (pp. 150–159). Morgan Kaufmann.

Dearden, R., Friedman, N., & Russell, S. (1998). Bayesian Q-learning. *Proceedings of Fifteenth National Conference on Artificial Intelligence (AAAI)* (pp. 761–768). AAAI Press.

Engel, Y., Mannor, S., & Meir, R. (2003). Bayes meets Bellman: the Gaussian process approach to temporal difference learning. *Proceedings of the Twentieth International Conference on Machine Learning (ICML)* (pp. 154–161).

[1]CECI is the 'Consortium des Equipements de Calcul Intensif'; a consortium of high-performance computing centers of UCL, ULB, ULg, UMons, and UNamur.

Engel, Y., Mannor, S., & Meir, R. (2005a). Reinforcement learning with Gaussian processes. *Proceedings of the Twentieth International Conference on Machine Learning (ICML)* (pp. 201–208).

Engel, Y., Szabo, P., & Volkinshtein, D. (2005b). Learning to control an octopus arm with Gaussian process temporal difference methods. *Proceedings of Advances in Neural Information Processing Systems (NIPS)* (pp. 347–354). MIT Press.

Fard, M. M., & Pineau, J. (2010). PAC-Bayesian model selection for reinforcement learning. *Neural Information Processing Systems (NIPS)*.

Ghavamzadeh, M., & Engel, Y. (2006). Bayesian policy gradient algorithms. *Proceedings of Advances in Neural Information Processing Systems (NIPS)*. MIT Press.

Ghavamzadeh, M., & Engel, Y. (2007). Bayesian actor-critic algorithms. *Proceedings of the Twenty-Fourth International Conference on Machine Learning (ICML)*.

Guez, A., Silver, D., & Dayan, P. (2012). Efficient Bayes-adaptive reinforcement learning using sample-based search. *Neural Information Processing Systems (NIPS)*.

Hennig, P., Stern, D., & Graepel, T. (2009). Bayesian quadratic reinforcement learning. *Neural Information Processing Systems (NIPS)*.

Kocsis, L., & Szepesvári, C. (2006). Bandit based Monte-Carlo planning. *European Conference on Machine Learning (ECML)*, 282–293.

Poupart, P. (2008). Model-based Bayesian reinforcement learning in partially observable domains. *International Symposium on Artificial Intelligence and Mathematics (ISAIM)*.

Ross, S., & Pineau, J. (2008). Model-based Bayesian reinforcement learning in large structured domains. *Proceedings of the Twenty-Fourth Conference on Uncertainty in Artificial Intelligence (UAI)*. AUAI Press.

Ross, S., Pineau, J., Chaib-draa, B., & Kreitmann, P. (2011). A Bayesian approach for learning and planning in partially observable Markov decision processes. *Journal of Machine Learning Research (JMLR)*, *12*, 1729–1770.

Strens, M. (2000). A Bayesian framework for reinforcement learning. *In Proceedings of the Seventeenth International Conference on Machine Learning (ICML)* (pp. 943–950). ICML.

A. OPPS settings

The OPPS implementation used in this paper differs from the one introduced in (Castronovo et al., 2012) by the set of variables used to build formulas. The set of variables considered in this paper is composed of three variables. Those three variables correspond to three Q-functions computed through value iteration, using three different models. Formally, given a set of transitions observed so-far, h, $N_h(x,u)$ the number of times a transition starting from the state-action pair (x,u) occurs in h, and $N_h(x,u,y)$ the number of times the transition (x,u,y) occurs in h, three transition functions are defined $f_{mean}, f_{uniform}, f_{self}$ as follows:

1. f_{mean} corresponds to the expectation of a Dirichlet posterior distribution computed from the current history and the chosen prior distribution. If θ_0 denotes the counters of the observed transitions of prior $p^0_{\mathcal{M}}()$, f_{mean} is defined as follows:

   $$\forall x,u,y : \theta^h_{x,u}(y) = \theta^0_{x,u}(y) + N_h(x,u,y)$$

   Formally, the mean transition model is defined as follows:

   $$\forall x,u,y : f_{mean}(x,u,y) = \frac{\theta^h_{x,u}(y)}{\sum_{y'} \theta^h_{x,u}(y')}$$

2. $f_{uniform}$ corresponds to the expectation of a Dirichlet posterior distribution computed from the current history and a uniform Dirichlet prior distribution. Formally, the uniform transition model is defined as follows:

   $$\forall x,u,y : f_{uniform}(x,u,y) = \frac{1 + N_h(x,u,y)}{|U| + N_h(x,u)}$$

3. f_{self} corresponds to the expectation of a Dirichlet posterior distribution computed from the current history and a counter initialization corresponding to a Dirac centred over a deterministic MDP where each state can only be reached from itself (for all actions). Formally, the self transition model is defined as follows:

   $$\forall x,u : f_{self}(x,u,x) = \frac{1 + N_h(x,u,x)}{1 + N_h(x,u)}$$

   $$\forall x,u,y \neq x : f_{self}(x,u,x) = \frac{N_h(x,u,x)}{1 + N_h(x,u)}.$$

B. Erratum

There was a mistake in the experiments reported in the first version of the paper. It lead to an overestimation of the performances of the BAMCP algorithm.

The mistake came from the fact that the reward function used in the BAMCP algorithm was not taken equal to $R(x,u,y')$ but well to the expected value of this reward function, namely $R'(x,u)$:

$$R'(x,u) = \sum_{y' \in X} P(y'|x,u)\ R(x,u,y')$$

In such a context, the BAMCP algorithm works significantly better, probably because the function $R'(x,u)$ contains additional knowledge of the transition matrix.