Active Learning of MDP Models

Mauricio Araya-López, Olivier Buffet,
Vincent Thomas, and François Charpillet

Nancy Université / INRIA
LORIA – Campus Scientifique – BP 239
54506 Vandoeuvre-lès-Nancy Cedex – France
firstname.lastname@loria.fr

Abstract. We consider the active learning problem of inferring the transition model of a Markov Decision Process by acting and observing transitions. This is particularly useful when no reward function is *a priori* defined. Our proposal is to cast the active learning task as a utility maximization problem using Bayesian reinforcement learning with belief-dependent rewards. After presenting three possible performance criteria, we derive from them the belief-dependent rewards to be used in the decision-making process. As computing the optimal Bayesian value function is intractable for large horizons, we use a simple algorithm to approximately solve this optimization problem. Despite the sub-optimality of this technique, we show experimentally that our proposal is efficient in a number of domains.

1 Introduction

Learning in Markov Decision Processes (MDPs) is usually seen as a means to maximize the total utility for a given problem. Nevertheless, learning the transition model of an MDP independently of the utility function—if it exists—can be a very useful task in some domains. For example, this can be used for learning the transition model in a batch process, where in a first stage we are interested in choosing the good actions for optimizing the information gathering process, and afterwards in a second stage, we are interested in earning rewards [5]. Moreover, there are some cases where we do not have access to the utility function, such as models for simulations or model refinement, where we want only to learn a good model, no matter which task the model will be used for.

Learning stochastic MDP models is an easy task if an exploration policy is given. In this case, the history of transitions can be seen as the data, and the problem of finding the optimal parameters for the selected distribution over the models can be solved by using likelihood maximization.

Here, we are concerned with actively learning the transition model, what raises a control problem. This amounts to finding a policy that explores optimally an MDP in order to acquire the best distribution over possible models. This differs from active supervised learning, where any sample can be queried at any time. In our setting, a sequence of actions is needed to reach a specific state from which to acquire a new sample. This is a complex problem since one has to reason on an imperfect model in order to improve that same model.

To our knowledge, there is not much research in active learning for arbitrary stochastic MDP models [19]. Indeed, one of the few works in this domain is about learning Dynamic Bayesian Networks (DBNs) for representing factored MDPs [9], where the authors conclude that actively learning transition models is a challenging problem and new techniques are needed to address this problem properly.

Our proposal is to use the Bayesian Reinforcement Learning (BRL) machinery with belief-dependent rewards to solve this active learning task. First we cast the learning problem as a utility maximization problem by using rewards that depend on the belief that is being monitored. Then, we define some performance criteria to measure the quality of distributions produced by different policies. Using these criteria, we derive the belief-dependent rewards that will be used to find exploration policies. Due to the intractability of computing the optimal Bayesian value function, we solve this problem sub-optimally by using a simple myopic technique called EXPLOIT.

Belief-dependent rewards have been used as heuristic methods for POMDPs. For example, in coastal navigation [15], convergence is sped up by using reward shaping based on an information-based criterion. Moreover, POMDPs with belief-dependent rewards have been recently studied in [1], where classical POMDP algorithms are applied to this type of rewards only with little modifications. Unfortunately, these techniques cannot be applied, for the same reason why POMDP algorithms are not used for standard BRL: the special type of beliefs used are not suitable for these algorithms.

The remainder of the paper is organized as follows. In Section 2 we give a short review of BRL and the algorithms that have been proposed so far. Then, in Section 3 we introduce the methodology used to solve this active learning problem as a BRL problem with belief-dependent rewards, including the selected performance criteria and their respective derived rewards. In Section 4 we present the results of several experiments over some MDP models taken from the state of the art. Finally, in Section 5 we close with the conclusion and future work.

2 Background

2.1 Reinforcement Learning

A *Markov Decision Process* (MDP) [13] is defined by a tuple $\langle \mathcal{S}, \mathcal{A}, T, R \rangle$ where $\mathcal{S}$ is a finite set of system *states*, $\mathcal{A}$ is a finite set of possible *actions*, the *transition* function T indicates the probability to transition from one state s to another s' when some action a is performed: $T(s,a,s') = Pr(s'|s,a)$, and $R(s,a,s')$ is the instant scalar *reward* obtained during this transition. Reinforcement Learning (RL) [18] is the problem of finding an optimal decision policy—a mapping $\pi : \mathcal{S} \rightarrow \mathcal{A}$—when the model ($T$ and R) is unknown but while interacting with the system. A typical performance criterion is the expected return

$$V^{\pi}_{H}(s) = E_{\pi}\left[\sum_{t=0}^{H} R(S_t, A_t, S_{t+1}) | S_0 = s\right],$$

where H is the planning horizon[1]. Under an optimal policy, this state value function verifies the Bellman optimality equation [3] (for all $s \in \mathcal{S}$):

$$V^{*}_{H}(s) = \max_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} T(s, a, s') \left[R(s, a, s') + V^{*}_{H-1}(s')\right],$$

and computing this optimal value function allows to derive an optimal policy by behaving in a greedy manner, i.e., by picking actions in $\arg\max_{a \in \mathcal{A}} Q^{*}(s, a)$, where a state-action value function Q_{π} is defined as

$$Q^{\pi}_{H}(s, a) = \sum_{s' \in \mathcal{S}} T(s, a, s') \left[R(s, a, s') + V^{\pi}_{H-1}(s')\right].$$

Typical RL algorithms either (i) directly estimate the optimal state-action value function Q^{*} (model-free RL), or (ii) learn T and R to compute V^{*} or Q^{*} (model-based RL). Yet, in both cases, a major difficulty is to pick actions so as to make a compromise between exploiting the current knowledge and exploring to acquire more knowledge.

2.2 Model-based Bayesian Reinforcement Learning

We consider here *model-based Bayesian Reinforcement Learning* [17] (BRL), i.e., model-based RL where the knowledge about the model—now a random vector $\boldsymbol{b}$—is represented using a—generally structured—probability distribution over possible transition models. An initial distribution $Pr(\boldsymbol{b}_0)$ has to be specified, which is then updated using the Bayes rule after each new transition (s, a, s'):

$$Pr(\boldsymbol{b}_{t+1} | \boldsymbol{b}_0, h_{t+1}) = Pr(\boldsymbol{b}_{t+1} | \boldsymbol{b}_t, s_t, a_t, s^{t+1}) Pr(\boldsymbol{b}_t | \boldsymbol{b}_0, h_t),$$

where $h_t = s_0, a_0, \cdots, s_{t-1}, a_{t-1}, s_t$ is the state-action history until t. This random variable is usually known as the *belief* over the model, and therefore defines a belief-MDP with an infinite state space. Solving optimally this belief-MDP is intractable due to the increasing complexity along the planning horizon, but formulating the reinforcement learning problem using a Bayesian approach provides a sound way of dealing with the exploration-exploitation dilemma. Even though POMDP algorithms deal with belief-MDPs, one cannot directly benefit from classical POMDP algorithms because of the particular type of belief space. Other—offline or online—approximate approaches have therefore been introduced, allowing in a number of cases to prove theoretical properties. Several approaches and approximation techniques have been proposed for BRL and, as presented in [2], most approaches belong to one of the three following classes, from

[1] For simplicity, in this paper we are focused on undiscounted finite horizon problems. However, a similar technique can be applied to the discounted infinite horizon case.

the simplest to the most complex: *undirected approaches*, *myopic approaches* and *belief-lookahead approaches*.

Undirected approaches do not consider the uncertainty about the model to select the next action, and therefore do not reason about the possible gain of information. They often rely on picking random actions occasionally, e.g., using an ϵ-greedy or softmax exploration strategy, the computed Q-value being based on the average model. These algorithms usually converge to the optimal value function in the limit, but with no guarantee on the convergence speed.

Myopic approaches select the next action so as to reduce the uncertainty about the model. Some of them solve the current average MDP with an added exploration reward which favors transitions with lesser known models, as in R-MAX [4], BEB [10], or with variance based rewards [16]. Another approach, used in BOSS [2], is to solve, when the model has changed sufficiently, an optimistic estimate of the true MDP (obtained by merging multiple sampled models). For some of these algorithms, such as BOSS and BEB, there is a guarantee that, with high probability, the value function is close to some optimum (Bayesian or PAC-MDP) after a given number of samples. Yet, they may stop exploring after some time, preventing the convergence to the optimal value function.

Belief-lookahead approaches aim at optimally compromising between exploration and exploitation. One can indeed [7] reformulate BRL as the problem of solving a POMDP where the current state is a pair $\omega = (s, \boldsymbol{b})$, where s is the current observable state of the BRL and $\boldsymbol{b}$ is the belief on the hidden model. Each transition (s, a, s') is an observation that provides new information to include to $\boldsymbol{b}$. BEETLE [12] is one of the few such approaches, one reason for their rarity being their overwhelming computational requirements. Other option is to develop the tree of beliefs and use branch-and-bound to prune the infinite expansion [6].

2.3 Chosen Family of Probability Distributions

Among various possible representations for the belief $\boldsymbol{b}$ over the model, we use here one independent Dirichlet distribution per state-action pair. We denote one of them at time t by its sufficient statistic: a positive integer vector $\boldsymbol{\theta}^t_{s,a}$ where $\boldsymbol{\theta}^t_{s,a}(s')$ is the number of observations of transition (s, a, s'), including $\boldsymbol{\theta}^0_{s,a}(s')$ *a priori* observations. The complete belief state of the system can thus be written $\omega = (s, \boldsymbol{\theta})$, where $\boldsymbol{\theta} = \{\boldsymbol{\theta}_{s,a}, \forall s, a\}$. This is called a Belief-Augmented MDP (BAMDP), a special kind of belief-MDP where the belief-state is factored into the system state and the model. A triplet (s, a, s') leads to a Bayesian update of the model, $\boldsymbol{\theta}'$ differing from $\boldsymbol{\theta}$ only in that $\boldsymbol{\theta}'_{s,a}(s') = \boldsymbol{\theta}_{s,a}(s') + 1$. Moreover, due to the properties of Dirichlet distributions, the transition function of the BAMDP $T(\omega, a, \omega')$ is given by: $Pr(\omega'|\omega, a) = \frac{\boldsymbol{\theta}_{s,a}(s')}{\|\boldsymbol{\theta}_{s,a}\|_1}$.

To sum up, BRL transforms the problem of facing an unknown model into that of making decisions when the state contains unobserved system parameters. The problem of finding a sound compromise between exploration and exploitation becomes that of solving a BAMDP given an initial set of belief-parameters $\boldsymbol{\theta}^0$.

3 Active Learning of MDP models using BRL

In this paper we are interested in learning the hidden model of an MDP by observing state transitions online, under an active exploration strategy. From this arises a decision-making problem, where the best policy of actions must be selected in order to optimize the learning process. For a given policy, the learning process is straightforward using the Bayesian setting, because the likelihood maximization for the joint Dirichlet distribution corresponds to the sequential Bayes update of the $\boldsymbol{\theta}$ parameter described in Section 2.2. Therefore, the optimal policy will depend on the criterion used to compare two joint Dirichlet distributions produced from different policies. Among the possible options, we have selected three performance criteria that will be described in Section 3.2.

For finding the optimal policy, we can cast the active learning task as a BRL problem with *belief-dependent rewards*, where these rewards can be derived from the performance criterion. In other words, we extend the classical definition of BRL to rewards that depend on the $\boldsymbol{\theta}$ parameter, where the Bellman equation takes the form:

$$V_H(\boldsymbol{\theta}, s) = \max_a \left[\sum_{s'} Pr(s'|s,a,\boldsymbol{\theta})(\rho(\boldsymbol{\theta},a,\boldsymbol{\theta}') + V_{H-1}(\boldsymbol{\theta}',s')) \right], \tag{1}$$

with $\boldsymbol{\theta}'$ the posterior parameter vector after the Bayes update with (s,a,s'), and $\rho(\boldsymbol{\theta},a,\boldsymbol{\theta}') = \rho(s,a,s',\boldsymbol{\theta})$ the immediate belief-dependent reward. Within this formulation the problem of actively learning MDP models can be optimally solved using a dynamic programming technique. Yet, as in normal BRL, computing the exact value function is intractable because of the large branching factor of the tree expansion, so approximation techniques will be needed to address this problem.

3.1 Derived Rewards

In order to define the belief-dependent rewards needed for Equation 1, we will use the analytical expressions of the performance criteria to derive analytical expressions for immediate reward functions. As our problem has a finite horizon, one can say that the performance criteria could be used directly as a reward in the final step, whereas the rewards would be zero for the rest of the steps. Yet, this type of reward functions forces to develop the complete tree expansion in order to obtain non-zero rewards, which turns out to be extremely expensive for large horizons.

Therefore, we need a way of defining substantial immediate rewards at each step. As rewards are defined over a transition (s,a,s') and the current belief-parameter $\boldsymbol{\theta}$, we will use the Bayesian update for computing the performance difference between the current belief and the posterior belief. This is a standard reward shaping technique, which allows decomposing a potential function—here the performance criteria—in immediate rewards for each step, with the property of preserving the optimality of the generated policy [11].

Let $D(\boldsymbol{\theta}^t, \boldsymbol{\theta}^0)$ be a distance between the initial prior and the posterior parameters after t Bayes updates such that maximizing this distance amounts to maximizing the gain of information. From this we define the derived reward as follows,

$$\rho(s, a, s', \boldsymbol{\theta}^t) = D(\boldsymbol{\theta}^{t+1}, \boldsymbol{\theta}^0) - D(\boldsymbol{\theta}^t, \boldsymbol{\theta}^0),$$

where $\boldsymbol{\theta}^{t+1}$ is the set of parameters after the transition (s, a, s') from $\boldsymbol{\theta}^t$. Please recall that the Bayes update only modifies one state-action pair per update, meaning that only one state-action pair component of our distribution will change per update. This provides important simplifications in computing the performance of one transition.

In some cases, the performance criterion complies with the triangular *equality*, meaning that the derived rewards can be simply computed as

$$\rho(s, a, s', \boldsymbol{\theta}^t) = D(\boldsymbol{\theta}^t, \boldsymbol{\theta}^{t+1}), \tag{2}$$

removing the dependency from the initial prior.

3.2 Performance Criteria

Assuming that the real model is unknown, we must define a way to compare two distributions produced by different policies. As there is no *a priori* best criterion, we have selected three information-based criteria under the constraint that they can be computed analytically: the *variance difference*, the *entropy difference* and the *Bhattacharyya distance*.

Variance Difference The first criterion is based on the simple intuition that we are seeking those policies that produce low-variance distributions. The variance for the multivariate distribution over the models corresponds to a heavily sparse matrix of size $|\mathcal{S}|^2|\mathcal{A}| \times |\mathcal{S}|^2|\mathcal{A}|$, but here we will consider that the sum of marginal variances (the trace of the matrix) is enough as a metric. The variance of the i-th element of a Dirichlet distribution is given by $\sigma^2(X_i|\boldsymbol{\alpha}) = \frac{\boldsymbol{\alpha}_i(\|\boldsymbol{\alpha}\|_1 - \boldsymbol{\alpha}_i)}{\|\boldsymbol{\alpha}\|_1^2(\|\boldsymbol{\alpha}\|_1 + 1)}$. Then, we define the *variance difference* as follows,

$$D_V(\boldsymbol{\theta}^t, \boldsymbol{\theta}^0) = \sum_{s,a} \sum_{s'} (\sigma^2(X_{s'}|\boldsymbol{\theta}^0_{s,a}) - \sigma^2(X_{s'}|\boldsymbol{\theta}^t_{s,a})).$$

Entropy Difference An other common measure for probability distributions is the entropy, which measures the uncertainty of a random variable. Computing the uncertainty of beliefs seems to be a natural way of quantifying the quality of distributions. The entropy of a multivariate random variable distributed as a Dirichlet distribution with parameters $\boldsymbol{\alpha}$ is given by $H(\boldsymbol{\alpha}) = \log(B(\boldsymbol{\alpha})) + (\|\boldsymbol{\alpha}\|_1 - N)\psi(\|\boldsymbol{\alpha}\|_1) - \sum_{j=1}^{N}((\boldsymbol{\alpha}_j - 1)\psi(\boldsymbol{\alpha}_j)$, where $B(\cdot)$ is the generalized beta function, N is the dimensionality of the vector $\boldsymbol{\alpha}$, and $\psi(\cdot)$ is the digamma function. Then, we define the *entropy difference* as

$$D_H(\boldsymbol{\theta}^t, \boldsymbol{\theta}^0) = \sum_{s,a} (H(\boldsymbol{\theta}^0_{s,a}) - H(\boldsymbol{\theta}^t_{s,a})).$$

Bhattacharyya Distance The two measures described above attempt to quantify how much information the distribution contains. In this context, information theory provides several notions of information such as Chernoff's, Shannon's, Fisher's or Kolmogorov's. As stated in [14], the last three are inappropriate for Dirichlet distributions, because an analytical solution of the integral does not exist or due to the non-existence for some specific values. Using the result in [14] for computing the Chernoff information between two Dirichlet distributions $C_\lambda(\boldsymbol{\alpha}, \boldsymbol{\alpha}')$, and fixing the parameter λ to $1/2$, we obtain the *Bhattacharyya distance* between two belief-states as follows,

$$D_B(\boldsymbol{\theta}^t, \boldsymbol{\theta}^0) = \sum_{s,a} -\log\left(\frac{B(\frac{\boldsymbol{\theta}^t_{s,a}}{2} + \frac{\boldsymbol{\theta}^0_{s,a}}{2})}{\sqrt{B(\boldsymbol{\theta}^t_{s,a})B(\boldsymbol{\theta}^0_{s,a})}}\right).$$

3.3 From Criteria to Rewards

It can be shown that the first two criteria presented in Section 3.2 comply with the triangular *equality*, so we can use Equation 2 to compute their respective derived reward functions. Even though this is not always true for the *Bhattacharyya* distance, we will also use Equation 2 for simplicity, knowing that we are not preserving optimality for this specific case.

Therefore, after some trivial but tedious algebra, we can define the *variance*, *entropy* and *Bhattacharyya* instant rewards. For presenting these expressions we use the helper variables $x = \boldsymbol{\theta}_{s,a}(s')$ and $y = \|\boldsymbol{\theta}_{s,a}\|_1$ for all the rewards, and also $z = \|\boldsymbol{\theta}_{s,a}\|_2^2$ for the variance reward:

$$\begin{aligned}
\rho_V(s,a,s',\boldsymbol{\theta}) &= \frac{1}{y+1} - \frac{z}{y^2(y+1)} + \frac{2x - y + z}{(y+1)^2(y+2)}, \\
\rho_H(s,a,s',\boldsymbol{\theta}) &= \log\left(\frac{y}{x}\right) + \frac{|\mathcal{S}|+1}{y} - \sum_{j=x}^{y} \frac{1}{j}, \\
\rho_B(s,a,s',\boldsymbol{\theta}) &= \log\left[\frac{\Gamma(x)\sqrt{x}}{\Gamma(x+1/2)}\right] - \log\left[\frac{\Gamma(y)\sqrt{y}}{\Gamma(y+1/2)}\right].
\end{aligned}$$

Also, we would like to introduce a simple reward function, motivated by the exploration bonus of BEB [10], which only focuses on the difference of information from one state-action pair to another. This *state-action count* reward can be simply defined as

$$\rho_S(s,a,s',\boldsymbol{\theta}) = \frac{1}{\|\boldsymbol{\theta}_{s,a}\|_1} = \frac{1}{y}.$$

This reward is easier to compute than the other three, but preserves the same principle of quantifying the information gain of a Bayes update. In fact, this reward function optimizes the performance criterion

$$D_S(\boldsymbol{\theta}^t, \boldsymbol{\theta}^0) = \sum_{s,a} (\psi(\|\boldsymbol{\theta}^t_{s,a}\|) - \psi(\|\boldsymbol{\theta}^0_{s,a}\|)),$$

which turns out to be a quantity appearing in both the *entropy difference* and the *Bhattacharyya distance*.

A key property of the rewards presented above is that they tend to zero for an infinite belief evolution, meaning that there is no more to learn at this stage. Specifically, the two last rewards ρ_B and ρ_S are always positive and decreasing functions with the belief evolution, while the two first ρ_V and ρ_H can have negative values, but their absolute values are always decreasing, all of them converging to zero in the limit.

3.4 Solving BRL with Belief-dependent Rewards

It is clear that the algorithms that have been introduced in Section 2.2 will require some modifications to work with belief-dependent rewards. For example, BEETLE uses a polynomial representation of the value function, where rewards are scalar factors multiplying monomials. In our setup, rewards will be functions multiplying monomials, which makes offline planning even more complex.

EXPLOIT, which is one of the simplest online algorithms for BRL, consists in solving the MDP corresponding to the current average model or, in other words, iterating over the Bayesian value function without performing the Bayes update. Then, EXPLOIT executes the best action for this simple MDP, updates its belief by observing the arrival state, and starts again by solving the MDP for the new average model. Please note that EXPLOIT does not guarantee converging to an optimal policy (as in ϵ-greedy Q-learning).

For the belief-dependent reward scenario, EXPLOIT takes the form

$$V_H(\boldsymbol{\theta}, s) = \max_a \left[\sum_{s'} Pr(s'|s, a, \boldsymbol{\theta}) \left(\rho(s, a, s', \boldsymbol{\theta}) + V_{H-1}(\boldsymbol{\theta}, s') \right) \right],$$

where the MDP to solve is defined by a transition model $T(s, a, s') = \boldsymbol{\theta}_{s,a}(s')/\|\boldsymbol{\theta}_{s,a}\|$ and reward function $R(s, a, s') = \rho(s, a, s', \boldsymbol{\theta})$.

If we consider now the belief-dependent rewards presented in Section 3.1, solving this MDP will not provide a lower bound—neither an upper bound—of the Bayesian value function, but only an approximation. This simple algorithm will exploit the current information about the belief to explore parts of the model where the information is still weak, and despite the sub-optimality of the approximation, this algorithm exhibits a fair exploration behavior where all state-action pairs are visited infinitely often in the limit.

4 Experiments

4.1 Experimental Setup

We have selected three small MDP models to learn, taken from the BRL state of the art: the classic *Bandit* [8] problem, and the *Chain* and *Loop* problems [17]. For each problem, we have tested several transition models, varying the

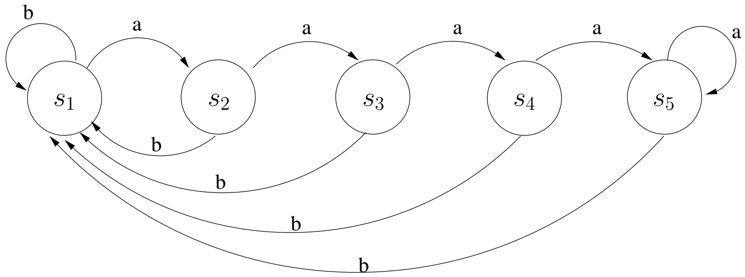

Fig. 1. Chain MDP model used for the experiments (without transition probabilities).

transition probabilities of the arcs to test our technique under different scenarios. However, due to space constraints we will present here only the results for the *Chain* problem, with a short discussion of the other problems.

In the 5-state **Chain MDP** (Figure 1), every state is connected to the state s_1 by taking action b and every state s_i is connected to the next state s_{i+1} with action a, except state s_5 that is connected to itself. In the *normal* case, the agent can "slip" at each time step with a probability of 0.2, performing the opposite action as intended. We have also tested the *deterministic* case when the probability of slipping is zero. Finally, we have tested a *mixed* version, where the probability of slipping is zero for action b, but 0.5 for action a. These two variations decrease the chances to arrive "by luck" to the states at the right of the chain.

The initial conditions of the problem are that we always start at state s_1 with the uniform distribution over the transition models as an initial prior. Other priors can be used—such as informative or structured priors—but for simplicity we will consider for this paper only the uniform one.

For evaluating the behavior of EXPLOIT, we have considered two other policies, namely the RANDOM and GREEDY policies. The RANDOM policy chooses homogeneously a random action at each step, while the GREEDY policy selects the action with largest expected immediate reward.

The three performance criteria have been tested, where the rewards for EXPLOIT and GREEDY are the respective derived rewards of Section 3.1 ρ_V, ρ_H and ρ_B depending on the evaluated performance criterion. Also, we have tested the *state-action count* reward ρ_S for each criterion and experiment.

For approximately solving the finite horizon MDPs within EXPLOIT, we have truncated the planning horizon to a small value $h = \min(2|\mathcal{S}|, H)$. Increasing h will provide better results, but to be similar in execution time with RANDOM and GREEDY, we have selected this small horizon.

4.2 Results

We have tested all the strategies on each problem for the first 100 to 1000 steps, and for each performance criterion. Figure 2 shows the average performance over

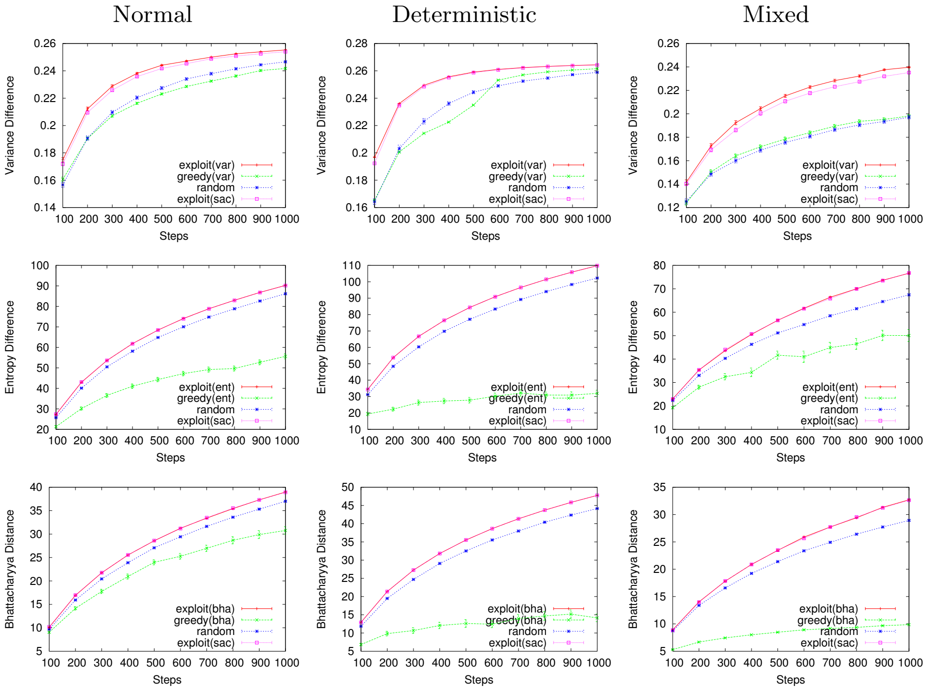

Fig. 2. Mean performance over 100 trials versus time steps, for the *Chain* problem with different models and the three performance criteria. For each plot, the RANDOM strategy (in blue $*$), the GREEDY strategy (in green $\times$), and the EXPLOIT algorithm with the derived reward (in red $+$) and with the *state-action count* reward (in magenta $\square$) are shown.

100 trials plotted with their respective 95% confidence interval for the *Chain* problem.

Even though *Chain* is a small problem, it is interesting because an intelligent exploration behavior is needed to learn the model. This can be seen along the three criteria, where the GREEDY policy behaves very poorly because a lookahead strategy is needed to arrive to some states. Even though the RANDOM strategy behaves fairly well, information-based strategies outperform this simple technique in all three criteria: for a desired solution quality, several hundred steps more are needed by RANDOM to achieve the same results. Even more, for the variance criterion, it seems to be very difficult for RANDOM to achieve the same quality in the short-term.

An other important result is that the *state-action count* reward behaves similarly well as the respective derived rewards. This means that the derived rewards can be replaced by this computationally cheap reward with no much performance loss. Indeed, performing a cross-experiment for the rewards and criteria shows that all information-based rewards behave similarly well for all criteria.

For the *Bandit* problem there is not much difference between the algorithms, and the behavior is the same through the different criteria. This is because the optimal policy for exploring a fully connected MDP corresponds to fairly selecting the available actions, which resembles the RANDOM policy.

On the *Loop* problem the results resemble to the ones presented for the *Chain*: information-based rewards outperform the two simple algorithms. Yet, the improvements of our approach compared to RANDOM are milder than in the *Chain*, because a simple exploration strategy is sufficient for this problem.

5 Conclusion and Future Work

We have presented a sound and original way of modeling the problem of actively learning a stochastic MDP model with arbitrary dynamics, by casting the problem as a BRL utility maximization problem with belief-dependent rewards. To that end, we have employed three performance criteria that are commonly used to compare probability distributions, namely the variance, the entropy and the Bhattacharyya distance. For each performance criterion, we have derived a belief-dependent reward such that, in the first two cases, the accumulated rewards correspond exactly to the performance criterion. Also, we have presented a simple reward function—the *state-action count*—based on previous work on normal BRL. Even though the formulation—in theory—allows solving the problem optimally, the intractability of computing the optimal Bayesian value function leads to using sub-optimal algorithms such as EXPLOIT. Our experiments show that this simple technique produces better results than selecting actions randomly, which is the baseline technique for exploring unknown MDP models. Also, our experiments show that there is no need for selecting complex derived rewards (at least for EXPLOIT) in order to obtain good results; the *state-action count* behaves nearly as well as the theoretically derived rewards.

However, this work leaves several open questions about the possibilities of modeling the active learning of MDP models using BRL. For instance, deepening the analysis on the relationship between the *state-action count* criterion and the other criteria might help defining a more advanced reward shaping technique to derive computationally inexpensive rewards. Also, exploring other techniques used for normal BRL could improve the results as they approach the optimal solution. For example, belief-lookahead techniques can be used for refining the myopic policies proposed here, or maybe some other myopic technique could produce better results.

A natural extension is to encode prior knowledge as a structured prior, such as with DBNs in [9], or with parameter tying in [12]. This would dramatically speed up the learning process by making a much more efficient use of data, while not involving major modifications in the solution techniques.

References

1. Araya-López, M., Buffet, O., Thomas, V., Charpillet, F.: A POMDP extension with belief-dependent rewards. In: Advances in Neural Information Processing Systems 23 (NIPS-10) (2010)
2. Asmuth, J., Li, L., Littman, M., Nouri, A., Wingate, D.: A Bayesian sampling approach to exploration in reinforcement learning. In: Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence (UAI'09) (2009)
3. Bellman, R.: The theory of dynamic programming. Bull. Amer. Math. Soc. 60, 503–516 (1954)
4. Brafman, R., Tennenholtz, M.: R-max - a general polynomial time algorithm for near-optimal reinforcement learning. Journal of Machine Learning Research 3, 213–231 (2003)
5. Şimşek, O., Barto, A.G.: An intrinsic reward mechanism for efficient exploration. In: Proceedings of the 23rd international conference on Machine learning. pp. 833–840. ICML'06, ACM, New York, NY, USA (2006)
6. Dimitrakakis, C.: Tree exploration for Bayesian RL exploration. In: CIMCA/IAWTIC/ISE. pp. 1029–1034 (2008)
7. Duff, M.: Optimal learning: Computational procedures for Bayes-adaptive Markov decision processes. Ph.D. thesis, University of Massachusetts Amherst (2002)
8. Gittins, J.C.: Bandit processes and dynamic allocation indices. Journal of the Royal Statistical Society 41(2), 148–177 (1979)
9. Jonsson, A., Barto, A.: Active learning of dynamic Bayesian networks in Markov decision processes. In: Proceedings of the 7th International Conference on Abstraction, Reformulation, and Approximation. pp. 273–284. SARA'07, Springer-Verlag, Berlin, Heidelberg (2007)
10. Kolter, J., Ng, A.: Near-Bayesian exploration in polynomial time. In: Proceedings of the Twenty-Sixth International Conference on Machine Learning (ICML'09) (2009)
11. Ng, A.Y., Harada, D., Russell, S.: Policy invariance under reward transformations: Theory and application to reward shaping. In: Proceedings of the Sixteenth International Conference on Machine Learning. pp. 278–287. Morgan Kaufmann (1999)
12. Poupart, P., Vlassis, N., Hoey, J., Regan, K.: An analytic solution to discrete Bayesian reinforcement learning. In: Proceedings of the Twenty-Third International Conference on Machine Learning (ICML'06) (2006)
13. Puterman, M.: Markov Decision Processes: Discrete Stochastic Dynamic Programming. Wiley-Interscience (April 1994)
14. Rauber, T., Braun, T., Berns, K.: Probabilistic distance measures of the Dirichlet and Beta distributions. Pattern Recognition 41(2), 637–645 (2008)
15. Roy, N., Thrun, S.: Coastal navigation with mobile robots. In: Advances in Neural Information Processing Systems 12. pp. 1043–1049 (1999)
16. Sorg, J., Singh, S., Lewis, R.: Variance-based rewards for approximate Bayesian reinforcement learning. In: Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence (2010)
17. Strens, M.J.A.: A Bayesian framework for reinforcement learning. In: Proceedings of the International Conference on Machine Learning (ICML'00). pp. 943–950 (2000)
18. Sutton, R., Barto, A.: Reinforcement Learning: An Introduction. MIT Press (1998)
19. Szepesvári, C.: Reinforcement learning algorithms for MDPs – a survey. Tech. Rep. TR09-13, Department of Computing Science, University of Alberta (2009)