Learning Exploration Policies with Models

Conference on Automated Learning and Discovery (CONALD'98)

Marco Wiering marco@idsia.ch
Jürgen Schmidhuber juergen@idsia.ch
IDSIA, Corso Elvezia 36
CH-6900 Lugano, Switzerland

Abstract

Reinforcement learning can greatly profit from world models updated by experience and used for computing policies. Fast discovery of near-optimal policies, however, requires to focus on "useful" experiences. Using an additional exploration model, we learn an exploration policy maximizing "exploration rewards" for visits of states that promise information gain. We augment this approach by an extension of Kaelbling's Interval Estimation algorithm to the model-based case. Experimental results in stochastic environments demonstrate advantages of this hybrid approach.

1 Introduction

Since a reinforcement learning (RL) agent is only able to learn from what it has experienced, the success of its computed policy heavily depends on the utility of its experiences. In RL the problem of selecting actions for information gain is called exploration or dual control (Dayan and Sejnowski, 1996).

Previous work. *Undirected* exploration methods use randomized action selection methods to guess useful experiences. *Directed* exploration methods learn an *exploration value function* in the same way standard RL methods learn a problem-oriented value function. They simply define an exploration reward function determining immediate exploration rewards, and let the selected RL method learn *exploration values*. Previous methods use Q-learning for learning where to explore (e.g., Schmidhuber 1991; Thrun 1992; Storck, Hochreiter and Schmidhuber, 1995). This can work significantly better than undirected methods. More recent research, however, also shows how undirected exploration techniques can be improved by using the action-penalty rule (Koenig and Simmons, 1996) which makes unexplored actions look more promising — this decreases the advantage of directed exploration. Another exploration strategy is embodied by the Interval Estimation (IE) algorithm (Kaelbling, 1993)

which uses second order statistics to detect whether certain actions have a potential of belonging to the optimal policy. IE computes confidence intervals of Q-values and always selects the action with largest upper interval boundary.

Our approach. We extend previous work by using model-based RL (MBRL) to learn exploration policies. Since MBRL can outperform its direct RL counterpart (Moore and Atkeson, 1993), we expect that it can also improve learning to explore. We will use a slightly adapted version of prioritized sweeping (PS) (Moore and Atkeson, 1993) to learn both an exploration policy and a problem-oriented policy, and combine this approach with (a) frequency-based and recency-based exploration reward functions, and (b) our novel Model-Based Interval Estimation (MBIE) update rule, which combines IE and MBRL.

Outline. Section 2 briefly describes MBRL. Section 3 addresses exploration in RL and mentions the exploration reward rules used in the experiments. Section 4 introduces MBIE. Section 5 describes experimental results on a 50×50 maze with one optimal goal and two suboptimal ones. Section 6 concludes.

2 Model-Based Reinforcement Learning

Inducing a model from experiences can simply be done by counting the frequency of observed experiences. Towards this end the agent uses the following variables:

$C_{ij}(a)$:= nr. of transitions from state i to j after executing action a.

$C_i\ (a)$:= number of times the agent has executed action a in state i.

$R_{ij}(a)$:= sum over all immediate rewards received after executing action a in state i and stepping to state j.

A maximum likelihood model (MLM) is computed as follows (where $\frac{0}{0} := 0$):

$$\hat{P}_{ij}(a) := \frac{C_{ij}(a)}{C_i(a)} \text{ and } \hat{R}(i,a,j) := \frac{R_{ij}(a)}{C_{ij}(a)} \tag{1}$$

Prioritized Sweeping (PS). Dynamic programming (DP) techniques could immediately be applied to the estimated model, but online DP tends to be computationally very expensive. To speed up DP algorithms, some sort of efficient update-step management should be performed. This can be done by prioritized sweeping (PS) (Moore and Atkeson, 1993) which assigns priorities to updating the Q-values of various state/action pairs according to a heuristic estimate of the update sizes.

Our PS. Moore and Atkeson's PS (M+A's PS) calculates priorities based on the largest single update of a successor state. It inserts states in the priority queue before their Q-values are updated. Our variant, however, updates Q-values of states before the states are inserted. This allows for computing the *exact* size of updates of state values since they have been used for updating the Q-values of their predecessors. Unlike our PS, M+A's PS cannot detect large

state-value changes due to many small update steps, and will forget to process the corresponding states.

3 Exploration

Max-Random exploration rule. Undirected exploration methods rely on pseudo-random generators, e.g., the *Boltzmann* exploration rules. We will use Max-Random, however, because it often outperforms Boltzmann (Thrun, 1992; Caironi and Dorigo, 1994). It uses a single parameter P_{max} denoting the probability of selecting the action with highest Q-value, and selects a random action otherwise.

Directed Exploration. Directed exploration techniques direct the exploration behavior to the most interesting parts of the state space. All they require is a local reward function determining which experience is interesting (e.g. Schmidhuber, 1991). It takes the place of the standard MDP reward function. The MDP transitions and the experiences stay the same, but now we learn two Q-functions: the *exploration Q-function* and the *exploitation Q-function.*

Recency-based. One of our local reward functions for exploring state/action pair (s_t, a_t) is: $R^E(s_t, a_t, s_{t+1}) := \frac{-t}{K_T}$, where K_T is a scaling constant and t the current time step.

Frequency-based. The other is: $R^E(s_t, a_t, s_{t+1}) := -\frac{C_{s_t}(a_t)}{K_C}$, where K_C is a scaling constant.

Learning exploration models. We use prioritized sweeping to quickly learn exploration models useful for learning Q-values estimating global information gain, taking into account yet unexplored regions of the state-space.

Replacing reward. To focus on the latest available information, we replace the estimated reward $\hat{R}(i, a, j)$ by $R^E(i, a, s_{t+1})$ for all j with $\hat{P}_{ij}(a) > 0$ in our exploration model.

4 Model Based Interval Estimation

To explore efficiently, an agent should not repeatedly try out actions that certainly cannot belong to the optimal policy. To reduce the set of optimal action candidates we extend the interval estimation algorithm (IE) (Kaelbling, 1993) to make it suitable to MBRL. IE selects the action with the largest upper bound for its Q-value. To compute upper bounds it keeps track of the means and standard deviations of all Q-values.

MBIE uses the model to compute the upper bound of Q-values. Given a set of outgoing transitions from state/action pair (i, a), MBIE increases the probability of the best transition (the one which maximizes $\gamma V(j) + R(i, a, j)$), depending on its standard deviation. Then MBIE renormalizes the transition

probabilities and uses the result for computing the Q-values. The following algorithm can be used in the prioritized sweeping algorithm:

Model Based Interval Estimation:

1) $m \leftarrow Argmax_{j:\hat{P}_{ij}(a)>0}\{R(i,a,j)+\gamma V(j)\}$

2) $P^+_{im}(a) \leftarrow \frac{\hat{P}_{im}(a)+\frac{z^2_\alpha}{2C_i(a)}+\frac{z_\alpha}{\sqrt{C_i(a)}}\sqrt{\hat{P}_{im}(a)(1-\hat{P}_{im}(a))+\frac{z^2_\alpha}{4C_i(a)}}}{1+\frac{z^2_\alpha}{C_i(a)}}$

3) $\Delta_P \leftarrow P^+_{im}(a) - \hat{P}_{im}(a)$

4) $\forall j \neq m$ 4.1) $P^+_{ij}(a) \leftarrow \hat{P}_{ij}(a) - \frac{\Delta_P C_{ij}(a)}{C_i(a)-C_{im}(a)}$

5 $Q(i,a) := \sum_j P^+_{ij}(a)(\hat{R}(i,a,j)+\gamma V(j))$

Here z_α is a variable which determines the confidence bounds — see (Kaelbling, 1993) for details.

MBIE hybrids. Although IE seems promising it does not clearly outperform Q-learning with Boltzmann exploration due to problems of estimating the variance of a changing Q-function in the beginning of the learning phase (Kaelbling, 1993). Since MBIE also relies on initial statistics we propose to start out with some other exploration method and switch to IE once some appropriate condition holds. After switching, we first copy the exploitation model and then we apply asynchronous value iteration (Bellman, 1961) to it; the iteration procedure calls MBIE for computing Q-values and ends once the maximal change of some state value is below some threshold.

5 Experiments

The problem. We use a 50×50 maze shown in Figure 1. It consists of about 20% blocked states and 20% penalty states (these are inserted randomly). In each state the agent can select one of four actions: *go north, go east, go south, go west.* There is a fixed starting state (S). There are three absorbing goal states, two of them are suboptimal (F), and one is optimal (G). Selected actions are replaced by random actions with 10% probability.

Reward function. Actions leading to a blocked state are not executed and punished by a reward of -2. Steps leading to free (penalty) states are punished by a reward of -1 (-10). If the agent finds the optimal (suboptimal) goal state it will receive a reward of 1000 (500). The discount factor γ is 0.99.

Comparison. We compare the following exploration methods: Max-Random, directed model-based exploration techniques using frequency-based and recency-based reward rules, and MBIE. The latter starts out with model-based exploration using the frequency-based reward rule, and switches to IE once the value function hardly changes any more (by less than 0.02 % per update).

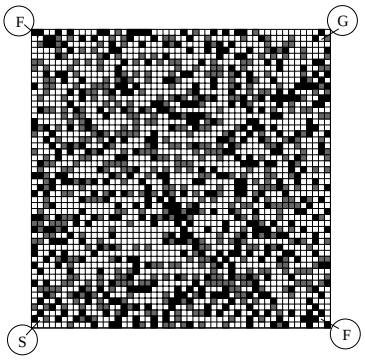

Figure 1: *The* 50×50 *maze used in the experiments. Black squares denote blocked fields, grey squares penalty fields. The agent starts at S and searches for a minimally punishing path to the optimal goal G. Good exploration is required to avoid focusing on suboptimal goal states (F).*

The goal is to learn good policies as quickly as possible. We computed an optimal policy using value iteration (Bellman, 1961) and tested this optimal policy by testing it for 1000 steps. Average reinforcement intake during 10,000 tests was $7590 \pm 2 = 7.59$K. For each method we conduct 20 runs of 100,000 learning steps. During each run we measure how quickly and how often the agent's policy collects 95%, 99% and 99.8% of what the optimal policy collects. This is done by averaging the results of 1000 test runs conducted every 2000 learning steps — each test run consists of executing the greedy policies.

Exploration Rule	95% (freq)	99% (freq)	99.8% (freq)	Best run result
Max-Random 0.2	43K (4)	52K (4)	68K (4)	4.8K ± 1.4K
Max-Random 0.4	— (0)	— (0)	— (0)	4.1K ± 0.3K
Frequency-based	24K (20)	50K (16)	66K (10)	7.55K ± 0.06K
Recency-based	30K (19)	51K (7)	79K (3)	7.3K ± 0.7K
MBIE	25K (20)	42K (19)	66K (18)	7.57K ± 0.05K

Table 1: *The number of steps required by several exploration methods for obtaining* ϵ*-optimal policies (and how many runs found them). The rightmost column shows average and standard deviation of the best test result during a run.*

Results. Table 1 shows significant improvements achieved by learning an exploration model. The undirected exploration methods focus too much on suboptimal goals (which are closer and therefore easier to find). Exploration model-based learning, however, does favor paths leading to the optimal goal. Using the frequency-based reward rule by itself, the agent always finds the

optimal goal although it often fails to find 99.8% optimal policies. Switching to MBIE after some time (about 45,000 steps), however, further improves matters. This strategy finds optimal or near-optimal policies in 90% of the cases and results in the best final performance.

6 Discussion

Undirected exploration applied to tasks with multiple absorbing goal states faces major difficulties in finding the optimal one. Exploration models, however, allow for discovering good policies circumventing suboptimal goal states. We compared two types of exploration rewards: frequency-based and recency-based. Although frequency-based exploration works best in stationary environments, recency-based reward rules may make more sense in non-stationary ones.

Estimating the variance of the Q-values can save unnecessary resampling of many actions. For this reason we introduced MBIE, which combines Kaelbling's interval estimation (IE) algorithm (Kaelbling, 1993) and model-based reinforcement learning (MBRL). Since MBIE heavily relies on initial statistics, we switch it on only after an initial phase during which an exploration model is learned (according to, say, the frequency-based local exploration reward rule). To our knowledge, this approach is currently the most effective exploration method for maze problems.

References

Bellman, R. (1961). *Adaptive Control Processes.* Princeton University Press.

Caironi, P. V. C. and Dorigo, M. (1994). Training Q-agents. Technical Report IRIDIA-94-14, Université Libre de Bruxelles.

Dayan, P. and Sejnowski, T. J. (1996). Exploration bonuses and dual control. *Machine Learning*, 25:5–22.

Kaelbling, L. (1993). *Learning in Embedded Systems.* MIT Press.

Koenig, S. and Simmons, R. G. (1996). The effect of representation and knowledge on goal-directed exploration with reinforcement-learning algorithms. *Machine Learning*, 22:228–250.

Moore, A. and Atkeson, C. G. (1993). Prioritized sweeping: Reinforcement learning with less data and less time. *Machine Learning*, 13:103–130.

Schmidhuber, J. (1991). Curious model-building control systems. In *Proc. International Joint Conference on Neural Networks, Singapore*, volume 2, pages 1458–1463. IEEE.

Storck, J., Hochreiter, S., and Schmidhuber, J. (1995). Reinforcement driven information acquisition in nondeterministic environments. In *Proceedings of the International Conference on Artificial Neural Networks*, volume 2, pages 159–164. EC2 & Cie, Paris.

Thrun, S. (1992). Efficient exploration in reinforcement learning. Technical Report CMU-CS-92-102, Carnegie-Mellon University.