

# REINFORCEMENT DRIVEN INFORMATION ACQUISITION IN NON-DETERMINISTIC ENVIRONMENTS

Jürgen Schmidhuber, Jan Storck, Josef Hochreiter, TUM

In Proc. ICANN'95, vol. 2, pages 159-164. EC2 & CIE, Paris, 1995.

## Abstract

For an agent living in a non-deterministic Markov environment (NME), what is, in theory, the fastest way of acquiring information about its statistical properties? The answer is: To design “optimal” sequences of “experiments” by performing action sequences that maximize expected information gain. This notion is implemented by combining concepts from information theory and reinforcement learning. Experiments show that the resulting method, *reinforcement driven information acquisition*, can explore certain NMEs much faster than conventional random exploration.

**Keywords:** Exploration, reinforcement learning, Q-learning, information gain, maximum likelihood models, non-deterministic Markovian environments, reinforcement directed information acquisition.

## 1. INTRODUCTION

Efficient reinforcement learning requires to model the environment. What is an efficient strategy for acquiring a model of a non-deterministic Markov environment (NME)? *Reinforcement driven information acquisition (RDIA)*, the method described in this paper, extends previous work on “query learning” and “experimental design” (see e.g. [3] for an overview, see [1, 6, 4, 7, 2] for more recent contributions) and “active exploration”, e.g. [9, 8, 11]. The method combines the notion of information gain with the notion of reinforcement learning. The latter is used to devise exploration strategies that maximize the former. Experiments demonstrate significant advantages of RDIA.

**Basic set-up / Q-Learning.** An agent lives in a NME. At a given discrete time step  $t$ , the environment is in state  $S(t)$  (one of  $n$  possible states  $S_1, S_2, \dots, S_n$ ), and the agent executes action  $a(t)$  (one of  $m$  possible actions  $a_1, a_2, \dots, a_m$ ). This affects the environmental state: If  $S(t) = S_i$  and  $a(t) = a_j$ , then with probability  $p_{ijk}$ ,  $S(t+1) = S_k$ . At certain times  $t$ , there is reinforcement  $R(t)$ . At time  $t$ , the goal is to maximize the discounted sum of future reinforcement  $\sum_{k=0}^m \gamma^k R(t+k+1)$  (where  $0 < \gamma < 1$ ). We use Watkins’ Q-learning [12] for this purpose:  $Q(S, a)$  is the agent’s evaluation (initially zero) corresponding to the state/action pair  $(S, a)$ . The central loop of the algorithm is as follows:

1. Observe current state  $S(t)$ . Randomly choose  $p \in [0, \dots, 1]$ . If  $p \leq \mu \in [0, \dots, 1]$ , randomly pick  $a(t)$ . Otherwise pick  $a(t)$  such that  $Q(S(t), a(t))$  is maximal.
2. Execute  $a(t)$ , observe  $S(t+1)$  and  $D(t)$ .
- 3.

$$Q(S(t), a(t)) \leftarrow (1 - \alpha)Q(S(t), a(t)) + \alpha(D(t) + \gamma \max_b Q(S(t+1), b)),$$

where  $0 < \gamma < 1, 0 < \alpha < 1$ . Goto 1.

## 2. MODEL BUILDING WITH RDIA

Our agent’s task is to build a model of the transition probabilities  $p_{ijk}$ . The problem is studied in isolation from goal-directed reinforcement learning tasks: RDIA embodies a kind of “*unsupervised reinforcement learning*”. It extends recent previous work on “active exploration” (e.g. [9, 8, 11]). Previous approaches (1) were limited to deterministic environments (they did not address the general problem of learning a model of the statistical properties of a non-deterministic NME), and (2) were based on ad-hoc elements instead of building on concepts from information theory.

**Collecting ML estimates.** For each state/action pair (or experiment)  $(S_i, a_j)$ , the agent has a counter  $c_{ij}$  whose value at time  $t$ ,  $c_{ij}(t)$ , equals the number of the agent’s previous experiences with  $(S_i, a_j)$ . In addition, for each state/action pair  $(S_i, a_j)$ , there are  $n$  counters  $c_{ijk}$ ,  $k = 1 \dots n$ . The value of  $c_{ijk}$  at time  $t$ ,  $c_{ijk}(t)$ , equals the number of the agent’s previous experiences with

$(S_i, a_j)$ , where the next state was  $S_k$ . Note that  $c_{ij}(t) = \sum_k c_{ijk}(t)$ . At time  $t$ , if  $c_{ij}(t) > 0$ , then

$$p_{ijk}^*(t) = \frac{c_{ijk}(t)}{c_{ij}(t)}$$

denotes the agent's current unbiased estimate of  $p_{ijk}$ . If  $c_{ij}(t) = 0$ , then we define (somewhat arbitrarily)  $p_{ijk}^*(t) = 0$ . Note that, as a consequence, before the agent has conducted any experiments of the type  $(S_i, a_j)$ , the  $p_{ijk}^*$  do not satisfy the requirements of a probability distribution. For  $c_{ij}(t) > 0$ , the  $p_{ijk}^*(t)$  build a maximum likelihood model (consistent with the previous experiences of the agent) of the probabilities of the possible next states.

**Measuring information gain.** If the agent performs an experiment by executing action  $a(t) = a_j$  in state  $S(t) = S_i$ , and the new state is  $S(t+1) = S_k$ , then in general  $p_{ijk}^*(t)$  will be different from  $p_{ijk}^*(t+1)$ . By observing the outcome of the experiment, the agent has acquired a piece of information. To measure its progress, we compute the information theoretic difference between what the agent knew before the experiment, at time  $t$ , and what the agent knew after the experiment, at time  $t+1$ . One natural way of doing this is to use the Kullback-Leibler distance (or asymmetric divergence) between the probability distributions represented by the  $p_{ijk}^*(t)$  and  $p_{ijk}^*(t+1)$ . We define

$$D(t) = \left| \sum_k d_k(t) \right|, \quad (1)$$

where

$$d_k(t) = 0 \quad \text{if } p_{ijk}^*(t+1) = 0 \quad \text{or} \quad p_{ijk}^*(t) = 0;$$

$$d_k(t) = p_{ijk}^*(t+1) \ln \frac{p_{ijk}^*(t+1)}{p_{ijk}^*(t)} \quad \text{otherwise.}$$

A related (but less informative) measure of progress is the entropy difference of the probability distributions represented by the  $p_{ijk}^*(t)$  and  $p_{ijk}^*(t+1)$ ,

$$D(t) = \left| \sum_k p_{ijk}^*(t+1) \ln p_{ijk}^*(t+1) - \sum_k p_{ijk}^*(t) \ln p_{ijk}^*(t) \right| \quad (2)$$

for  $c_{ij}(t) > 0$ . Again, if  $c_{ij}(t) = 0$  (before the agent has conducted any experiments of type  $(S_i, a_j)$ ), the entropy of the corresponding MLM is taken to be zero. In this case,  $D(t)$  will be zero, too. Another simple, related performance measure is  $D(t) = \sum_k |p_{ijk}^*(t+1) - p_{ijk}^*(t)|$  for  $c_{ij}(t) > 0$ , and  $D(t) = 0$  for  $c_{ij}(t) = 0$ . Initial experiments seem to indicate that the particular definition of  $D(t)$  does not make an essential difference.

**In all cases, best policies are found by using  $D(t)$  as the reinforcement  $R(t)$  for the Q-Learning algorithm from section 2.** Since an experiment at time  $t$  affects only  $n$  estimates (the  $n$   $p_{ijk}^*(t+1)$  associated with  $a_j = a(t)$  and  $S_i = S(t)$ ), and since  $D(t)$  can always be computed within  $O(n)$  operations, the algorithm's overall complexity per time step is bounded by  $O(n)$ .

# Experiments	Random Search	RDIA (entropy)	RDIA (prob. diff.)
1	204.93	204.93	204.93
1024	2.97	67.73	65.49
2048	3.40	40.59	21.98
4096	2.74	10.57	5.30
8192	3.72	4.08	3.88
16384	4.11	2.44	2.30
32768	3.43	1.27	1.44
65536	2.03	0.76	0.88
131072	1.58	0.54	0.59
262144	1.07	0.35	0.35

Table 1: *For random search and two RDIA variants, the evolutions of the sum of Kullback-Leibler distances between estimated and true probability distributions are shown. In the beginning, RDIA takes a while to find out where it can expect to learn something. But then it quickly surpasses random search.*

### 3. SIMULATIONS OF RDIA

We compared the performance of several RDIA variants as described above to the performance of conventional random exploration (variants of random exploration are the methods employed by most authors).

**A small environment.** The first test environment consists of  $n = 10$  states. There are  $m = 10$  possible actions, and 100 possible experiments. The transition probabilities are:

$$\begin{aligned}
 p_{ijk} &= 1 \text{ for } i = 1, \dots, 9; j = 1, \dots, 9; k = i; \\
 p_{ijk} &= 1 \text{ for } i = 1, \dots, 9; j = 10; k = i + 1; \\
 p_{ijk} &= \frac{1}{10} \text{ for } i = 10; j = 1, \dots, 10; k = 1, \dots, 10;
 \end{aligned}$$

and  $p_{ijk} = 0$  otherwise. The only state that allows to acquire a lot of information is  $S_{10}$ . After a while, RDIA (with parameters  $\alpha = 0.5$ ,  $\gamma = 0.9$ , and  $\mu = 0.1$ ) discovers this and establishes a policy that causes the agent to move as quickly as possible to  $S_{10}$  from every other state. Random exploration, however, wastes most of the time on the soon useless (uninformative) examination of the states  $S_1 \dots S_9$ . This can be seen from table 1, which compares random search and the two RDIA variants that worked best: (1) RDIA based on changes in entropy (equation 2), (2) RDIA based on weighted probability changes. In the beginning, RDIA takes a while to find out where it can expect to learn something. **Then it quickly catches on and surpasses random search.**

**A bigger environment.** The second test environment consists of  $n = 100$  states. There are  $m = 100$  possible actions, and 10000 possible experiments.

Goal entropy	# Experiments: Random Search	#Experiments: RDIA
170.0	$3.0 * 10^6$	$1.1 * 10^6$
370.0	$2.9 * 10^7$	$2.5 * 10^6$
459.0	$1.6 * 10^9$	$2.6 * 10^7$
460.0	unknown	$6.8 * 10^7$

Table 2: For random search and for RDIA based on entropy differences, this table shows the number of time steps required to achieve given entropy values. The optimal value (the true information content of the environment) is 460.517019. As the entropy margin approaches the optimum, RDIA becomes much faster. The entry marked “unknown” was not computed due to limited computation time.

The transition probabilities are:

$$\begin{aligned}
 p_{ijk} &= 1 \text{ for } i = 1, \dots, 99; j = 1, \dots, 99; k = i; \\
 p_{ijk} &= 1 \text{ for } i = 1, \dots, 99; j = 100; k = i + 1; \\
 p_{ijk} &= \frac{1}{100} \text{ for } i = 100; j = 1, \dots, 100; k = 1, \dots, 100;
 \end{aligned}$$

and  $p_{ijk} = 0$  otherwise. The information content of the second environment (the sum of the entropies of the true transition probability distributions associated with all state/action pairs) is 460.517019.

For random search and for RDIA based on entropy changes (with parameters  $\alpha = 0.5$ ,  $\gamma = 0.9$ , and  $\mu = 0.1$ ), table 2 shows the number of time steps required to achieve given entropy values. The only state allowing for acquisition of a lot of information is  $S_{100}$ . RDIA quickly discovers this and establishes a policy that causes the agent to move as quickly as possible to  $S_{100}$  from every other state. Random exploration, in contrast, wastes much of its time on the states  $S_1 \dots S_{99}$ . Again, for small entropy margins, the advantage of reinforcement driven information acquisition is not as pronounced as in later stages, because Q-learning needs some time to fix the strategy for performing experiments. As the entropy margin approaches the optimum, however, reinforcement driven information acquisition becomes much faster, by at least an order of magnitude.

**Future work.** 1. “*Exploitation/exploration trade-off*”. In this paper, exploration was studied in isolation from exploitation. Is there an “optimal” way of combining both? For which kinds of *goal-directed* learning should RDIA be recommended? It is always possible to design environments where “curiosity” (the drive to explore the world) may “kill the cat”, or at least may have a negative influence on exploitation performance. This is illustrated by additional experiments presented in [10]: In one environment described therein, exploration helps to speed up exploitation. But with a different environment,

curiosity slows down exploitation. The “exploitation/exploration trade-off” remains an open problem.

2. *Additional experimental comparisons.* It will be interesting to compare RDIA to better competitors than random exploration, like e.g. Kaelbling’s Interval Estimation algorithm [5].

3. *Function approximators.* It also will be interesting to replace the Q-table by function approximators like backprop networks. Previous experimental work by various authors indicates that in certain environments this might improve performance, despite the fact that theoretical foundations of combinations of Q-learning and function approximators are still weak.

## References

- [1] E. B. Baum. Neural nets that learn in polynomial time from examples and queries. *IEEE Transactions on Neural Networks*, 2(1):5–19, 1991.
- [2] D. A. Cohn. Neural network exploration using optimal experiment design. In J. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems 6*. San Mateo, CA: Morgan Kaufmann, 1994.
- [3] V. V. Fedorov. *Theory of optimal experiments*. Academic Press, 1972.
- [4] J. Hwang, J. Choi, S. Oh, and R. J. Marks II. Query-based learning applied to partially trained multilayer perceptrons. *IEEE Transactions on Neural Networks*, 2(1):131–136, 1991.
- [5] L. Kaelbling. *Learning in Embedded Systems*. MIT Press, 1993.
- [6] D. J. C. MacKay. Information-based objective functions for active data selection. *Neural Computation*, 4(2):550–604, 1992.
- [7] M. Plutowski, G. Cottrell, and H. White. Learning Mackey-Glass from 25 examples, plus or minus 2. In J. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems 6*, pages 1135–1142. San Mateo, CA: Morgan Kaufmann, 1994.
- [8] J. H. Schmidhuber. Curious model-building control systems. In *Proc. International Joint Conference on Neural Networks, Singapore*, volume 2, pages 1458–1463. IEEE, 1991.
- [9] J. H. Schmidhuber. A possibility for implementing curiosity and boredom in model-building neural controllers. In J. A. Meyer and S. W. Wilson, editors, *Proc. of the International Conference on Simulation of Adaptive Behavior: From Animals to Animats*, pages 222–227. MIT Press/Bradford Books, 1991.

- [10] J. Storck. Reinforcement-Lernen und Modellbildung in nicht-deterministischen Umgebungen. Fortgeschrittenenpraktikum, Fakultät für Informatik, Lehrstuhl Prof. Brauer, Technische Universität München, 1994.
- [11] S. Thrun and K. Möller. Active exploration in dynamic environments. In D. S. Lippman, J. E. Moody, and D. S. Touretzky, editors, *Advances in Neural Information Processing Systems 4*, pages 531–538. San Mateo, CA: Morgan Kaufmann, 1992.
- [12] C. Watkins. *Learning from Delayed Rewards*. PhD thesis, King's College, 1989.