REINFORCEMENT DRIVEN INFORMATION ACQUISITION IN NONDETERMINISTIC ENVIRONMENTS

Jan Storck∗  Sepp Hochreiter†  Jürgen Schmidhuber‡
Fakultät für Informatik  Technische Universität München 80290 München, Germany
IDSIA  Corso Elvezia 36 6900 Lugano, Switzerland

Abstract

For an agent living in a nondeterministic 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, nondeterministic Markovian environments, reinforcement directed information acquisition.

INTRODUCTION

Efficient reinforcement learning requires to model the environment. What is an efficient strategy for acquiring a model of a nondeterministic Markov environment (NME)? Reinforcement driven information acquisition (RDIA), the method described in this paper (first presented in [11]), extends previous work on “query learning” and “experimental design” (see e.g. [4] for an overview, see [1, 7, 5, 8, 3] for more recent contributions) and “active exploration”, e.g. [10, 9, 13]. 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 in certain NMEs.

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 \( S_1, S_2, ..., S_n \)), and the agent executes action \( a(t) \) (one of \( m \) possible actions \( a_1, a_2, ..., a_m \)). 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 [14] for

∗storck@informatik.tu-muenchen.de
†hochreit@informatik.tu-muenchen.de
‡juergen@idsia.ch - Proc. ICANN’95, vol. 2, pages 159-164. EC2 & CIE, Paris, 1995
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 $p \in [0, 1]$. If $p \leq \mu \in [0, 1]$, 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 $R(t)$.

3. $Q(S(t), a(t)) \leftarrow (1 - \beta)Q(S(t), a(t)) + \beta(R(t) + \gamma \max_b Q(S(t + 1), b)),$

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

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. [10, 9, 13]). 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 nondeterministic 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 $c_{ijk}$, $k = 1...n$. 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 increasing the estimators’ accuracy. In what follows, we will list three related variants of quantifying the agent’s progress. In all three cases, the agent’s progress will be taken as its reinforcement.

1. Standard statistics tells us that the approximative confidence region of a multinomial distribution satisfies $P(E^2(p_{ijk}^*, p_{ijk}) < \frac{\chi^2_{n-1} - \alpha}{c_{ij}}) = 1 - \alpha$, where $\alpha$ is a given confidence level (e.g. [2], p. 301), and $E^2(p_{ijk}^*, p_{ijk}) = \sum_{k=1}^n \frac{(p_{ijk}^* - p_{ijk})^2}{p_{ijk}}$ is the $p_{ijk}^*$ estimators’ weighted squared error (Pearson’s $\chi^2$-distance for multinomial distributions is a standard way of measuring the estimators’ deviations from the true probabilities, e.g. [2], p. 233, 301). Note that with confidence level $\alpha$, $E^2$’s upper bound is $\chi^2_{n-1, \alpha}$ (the $\alpha$-quantile of the $\chi^2_{n-1}$-distribution, which is independent of $c_{ij}$) divided by $c_{ij}$. This upper bound is proportional to $\frac{1}{c_{ij}}$. Decreasing $E^2$’s upper bound reduces the dispersions of the estimators, which is a central goal of optimal experiment design (e.g. [3, 4]). Hence, when it comes to choosing a new experiment, state/action pairs with small counters $c_{ij}$ are preferable. However, in the case of partly deterministic environments it would be smarter to conduct less “deterministic” experiments than nondeterministic ones. To take this additional aspect into account, the information gain (the agent’s current progress)
may be simply measured by
\[
D(t) = \sum_k |p_{ijk}^*(t+1) - p_{ijk}^*(t)|
\]
for \(c_{ij}(t) > 0\), and \(D(t) = 0\) for \(c_{ij}(t) = 0\). Note that \(p_{ijk}^*(t+1) - p_{ijk}^*(t)\) is proportional to \(\frac{1}{c_{ij}}\) for large \(c_{ij}\), but (unlike \(\frac{1}{c_{ij}}\) itself) these differences are zero for deterministic state/action pairs.

2. Since the estimators represent probability distributions over the next states \(S_k\), we may measure the agent’s progress by measuring probability distribution changes, e.g. by using the entropy difference of the probability distributions represented by the \(p_{ijk}^*(t+1)\) values and the \(p_{ijk}^*(t)\) values. We may redefine:
\[
D(t) = |\sum_k p_{ijk}^*(t+1) \ln p_{ijk}^*(t+1) - \sum_k p_{ijk}^*(t) \ln p_{ijk}^*(t)|
\]
for \(c_{ij}(t) > 0\). (For \(p_{ijk}^* = 0\), we define \(p_{ijk}^* \ln p_{ijk}^* = 0\). 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. Again, it can be shown that \(D(t)\) is proportional to \(\frac{1}{c_{ij}}\) for large \(c_{ij}\) (and zero in the deterministic case). Probability distributions with high entropy cause high \(D(t)\), which is precisely what is desired: high entropy distributions should be explored more than low entropy distributions (bias towards “nondeterministic” state/action pairs).

3. A related way for measuring probability distribution changes is to use the Kullback-Leibler distance. We may redefine \(D(t) = \sum_k d_k(t)\), where \(d_k(t) = 0\) if \(p_{ijk}^*(t+1) = 0\) or \(p_{ijk}^*(t) = 0\), and \(d_k(t) = p_{ijk}^*(t+1) \ln \frac{p_{ijk}^*(t+1)}{p_{ijk}^*(t)}\) otherwise. This measure has properties similar to those of the entropy difference above but is more sensitive to increases of the largest \(p_{ijk}^*\) values (emphasis on changes of probability distributions tending towards determinism).

The clue is: in all cases, the agent’s progress \(D(t)\) is used as the reinforcement \(R(t)\) for the Q-Learning algorithm from the introduction. 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 \(O(n)\).

Since all three \(D(t)\) variants not only prefer nondeterminism but also are proportional to \(\frac{1}{c_{ij}}\) for large \(c_{ij}\), the particular definition of \(D(t)\) should not make an essential difference. This is confirmed by initial experiments (see next section).

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:

\[
p_{ijk} = 1 \quad for \quad i = 1, \ldots 9; \quad j = 1, \ldots 9; \quad k = i;
\]

\[
p_{ijk} = 1 \quad for \quad i = 1, \ldots 9; \quad j = 10; \quad k = i + 1;
\]

\[
p_{ijk} = \frac{1}{10} \quad for \quad i = 10; \quad j = 1, \ldots 10; \quad k = 1, \ldots 10;
\]

and \(p_{ijk} = 0\) otherwise. The only state that allows for acquiring a lot of information is \(S_{10}\). After a while, RDIA (with parameters \(\beta = 0.5, \gamma = 0.9, \mu = 0.1\)) discovers this and establishes
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.

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

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 $S_1$ ... $S_9$. This can be seen from table 1, which compares random search and the two RDIA variants that worked best: RDIA based on changes in entropy (equation (2)), RDIA based on probability changes (equation (1)). 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. The transition probabilities are:

\[
p_{ijk} = 1 \quad \text{for} \quad i = 1, \ldots, 99; \quad j = 1, \ldots, 99; \quad k = i;
\]

\[
p_{ijk} = 1 \quad \text{for} \quad i = 1, \ldots, 99; \quad j = 100; \quad k = i + 1;
\]

\[
p_{ijk} = \frac{1}{100} \quad \text{for} \quad i = 100; \quad j = 1, \ldots, 100; \quad k = 1, \ldots, 100;
\]

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 $\beta = 0.5$, $\gamma = 0.9$, $\mu = 0.1$), 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 $S_1$ ... $S_{99}$. 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 [12]: 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 [6].

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


