INTERNALLY DRIVEN Q-LEARNING
Convergence and Generalization Results

Eduardo Alonso[1], Esther Mondragón[2] and Niclas Kjäll-Ohlsson[1]

[1] Department of Computing, City University London, London EC1V 0HB, U.K.

[2] Division of Psychology and Language Sciences, University College London, London WC1H 0AP, U.K.

eduardo@soi.city.ac.uk, e.mondragon@ucl.ac.uk, niclasko@gmail.com

Keywords: Q-learning, IDQ-learning, Internal Drives, Convergence, Generalization.

Abstract: We present an approach to solving the reinforcement learning problem in which agents are provided with internal drives against which they evaluate the value of the states according to a similarity function. We extend Q-learning by substituting internally driven values for *ad hoc* rewards. The resulting algorithm, Internally Driven Q-learning (IDQ-learning), is experimentally proved to convergence to optimality and to generalize well. These results are preliminary yet encouraging: IDQ-learning is more psychologically plausible than Q-learning, and it devolves control and thus autonomy to agents that are otherwise at the mercy of the environment (i.e., of the designer).

1 INTRODUCTION

Traditionally, the *reinforcement learning problem* is presented as follows: An agent exists in an environment described by some set of possible states, where it can perform a number of actions. Each time it performs an action in some state the agent receives a real-valued reward that indicates the immediate value of this state-action transition. This generates a sequence of states, actions and immediate rewards. The agent's task is to learn a control policy, which maximizes the expected sum of rewards, typically with future rewards discounted exponentially by their delay (Sutton & Barto, 1998).

It is therefore a working premise that the agent does not know anything about the environment or itself. Rewards are dictated by the environment not part of the environment and thus defined separated from outcomes. As a consequence only estate-action values are learned. In addition, learning is completely depended on the reward structure: If the reward changes, a new policy has to be relearned.

Let's emphasize this point: The only information available to the agent about a state is the amount of reward it predicts; it does not know why the new state is good or bad. Thus the agent is completely dependent on the environment to provide the correct reward values in order to guide its behaviour. The agent has no way of reasoning about the states in terms of its internal needs, because it has no internal needs other than reward maximization.

In this paper, we propose to redefine the value of an outcome as a function of the agent's motivational state. Because different states can be of different importance to the agent it is the responsibility of the agent to encode its own state signal. This proposal contradicts reinforcement learning where the value of an outcome is provided explicitly and separately from the actual outcome in the form of a reward.

Allegedly the most popular reinforcement learning algorithm is Q-learning, an off-policy algorithm where the optimal expected long-term return is locally and immediately available for each state-action pair. A one-step-ahead search computes the long-term optimal actions without having to know anything about possible successor states and their values. Under certain assumptions, Q-learning has been proved to converge with probability 1 to the optimal policy (Watkins & Dayan, 1992). In large state spaces, Q-learning has been successfully combined with function approximators.

In the next section, a variation of Q-learning, the Internally Driven Q-learning algorithm (IDQ-learning henceforth), based on the idea described above is presented. We show that IDQ-learning converges to the optimal policy and that it generalizes well in subsequent sections.

2 IDQ-LEARNING

2.1 States

A state is formally defined as a vector of elements where each element represents some modality along with a value. Significantly, elements can be shared across states.

$$\begin{aligned} &S_i = \{s_1,...,s_n\} \\ &Mod.S_i = \{0,1,....,n\} \\ &VS_i = [0,\max - strength] \end{aligned} \tag{1}$$

2.2 Similarity Function

In order to compare states with internal drives and among themselves a similarity function is introduced. This function should return a value between 0 and 1 when comparing two states, where 0 is no similarity and 1 denotes equality. The Gaussian function is a good match for the purpose. Let the modality strength of state S_i represent the mean μ of a normal distribution ND with standard deviation σ, and let VS_j and VS_i represent the value of state S_j and S_i respectively. The probability of state S_i occurring in ND is given by:

$$P(VS_i) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(VS_i-\mu)^2}{2\sigma^2}} \tag{2}$$

The same applies to $P(VS_j)$. This leads to the definition of the similarity function:

$$sim(S_i,S_j) = \frac{P(VS_i)}{P(VS_j)} I(S_i,S_j) \tag{3}$$

where

$$I(S_i,S_j) = \begin{cases} 1 & if\ Mod.S_i = Mod.S_j \\ 0 & if\ Mod.S_i \neq Mod.S_j \end{cases} \tag{4}$$

Thus the similarity function is bell-shaped, continuous, and insensitive to the sign of the difference between the values of two states.

2.3 Internal Drives

For the purpose of deriving outcome values the internal drives (ID) of an agent are defined as a vector of states along with a category indicator for each element denoting whether the state is aversive or appetitive:

$$\begin{aligned} &ID_i = \{s_1,...,s_n\} \\ &Mod.S_i = \{0,1,....,n\} \\ &VS_i = [0,\max - strength] \\ &CatS_i = \{aversive, appetitive\} \\ &aversive = -1 \\ &appetitive = 1 \end{aligned} \tag{5}$$

The asymptotic value of an outcome is thus set in the following way:

$$\lambda(S_i) = \max\left(\forall S_j \in ID : sim(S_i,S_j)\right) \times Cat.S_j \tag{6}$$

The value of an outcome is the maximum similarity when compared to all internal drives. This departure from traditional reinforcement learning is significant since: (a) in our proposal, states have intrinsic values defined as their similitude with internal drives –they are not given by the environment in an *ad hoc* manner; (b) rewards are not defined on state-action pairs –they *define* the states. As a consequence, the agents are in control, they are now cognitive agents.

2.4 The IDQ-learning Algorithm

The IDQ-learning learning is similar to the Q-learning algorithm. Its pseudo code reads as follows:

```
1. Initialize $Q(s, a)$ according to $\lambda(s)$
2. Repeat (for each episode)
3. Initialize $s$
4. Repeat (for each step of the episode)
5. Choose $a$ from $s$ using a policy derived from $Q$
6. Take action $a$, observe $s'$
7. $Q(s,a) \leftarrow Q(s,a) + \alpha[\lambda(s) + \gamma \max_{a'} Q(s',a') - Q(s,a)]$
8. $s \longleftarrow s'$
9. until $s$ is terminal
```

The three main novelties refer to steps 1, 6 and 7, specifically: the initial Q value is not arbitrary (typically 0, for lack of any information about the states); since r is now a defining characteristic of s', namely $\lambda(s)$, in step 6 there is no need to observe r; accordingly in step 7, $\lambda(s)$ takes the place of r in Q-learning. Because states have been defined as compounds of elements, step 7 above applies to the summation of the values of corresponding elements, forming what we call the state's expectance memory (EM) –we haven't made it explicit in the pseudo code to avoid over-indexing.

This framework makes explicit the two main conditions for the transfer of information, contingency and contiguity: agents make predictions on the value of states to come but, unlike in Q-learning, such values as well as the values of immediate rewards are defined as their probability of occurrence (of the values *themselves* not of the states). Moreover, generalization follows directly: agents do not need to have experienced previously a state in order to value it. As long as it shares elements with an ID or with a previously experienced state, it inherits a value.

3 EXPERIMENTS

In the next sub-sections we show experimentally how IDQ-learning converges to an optimal policy and how it generalizes in a traditional Grid-world domain.

The following parameters were set for the IDQ agent: σ = 0.2 (the sensitivity of the similarity function), α = 0.1 (learning rate of CS), ε = 0.8 (choose the greedy response in 80% of the cases), and γ = 0.9 (reward discounting).

3.1 Convergence

Convergence is measured by recording the average absolute fluctuation (AAF) for expectance memory per episode. This is done by accumulating the absolute differences $diff_{abs}$ between the values of the expectance memory and their values after an update has been carried out. Additionally the number of steps to reach the goal, $episode_{length}$, is recorded. The AAF is thus given by $diff_{abs}$ / $episode_{length}$. For IDQ there can be several updates of the expectance memory. This is because of the consideration of states as compounds, where each element of the compound enters into separate associations from the others. It is therefore necessary to record the number of updates per episode step $numerrors_{episodestep}$ as well. The average absolute fluctuation per episode for IDQ is given by $(diff_{abs}/numerrors_{episodestep})/episode_{length}$.

The optimal policy is the shortest path from any spatial location to the goal. Table 1 shows a summary of results for convergence experiments, where MPC stands for Maximum Policy Cost for learning period and OFPE stands for Optimal Policy Found after *n* Epochs.

Table 1: Convergence: MPC and OPFE per Grid type.

Grid	MPC	OPFE
3×3	13	15
5×5	71	13
10×10	626	843

As expected, the algorithm converges to the optimal results.

3.2 Generalization

Generalization in the Grid-world means how the consideration of states as compounds can help the transfer of learning between similar situations. When an element *X* at two different locations signals the same outcome, there is said to be a sharing of associations between the two locations. In Figure 1 element *X* enters into an association with the outcome state *G*. This association is strengthened at two locations. Additionally, both elements *A* and *B* enter into an association with *G* separately. In this situation it is said that element *X* is generalized from location (2,3) to location (3,2) and vice versa. There is thus a sharing of element *X* between the states at location (2,3) and at location (3,2). An algorithm which manages to gain a savings effect from this type of sharing is said to be able to generalize. This generalization and sharing effect should be manifested in faster convergence to the optimal policy if the algorithm is successful in using the redundant association to its benefit.

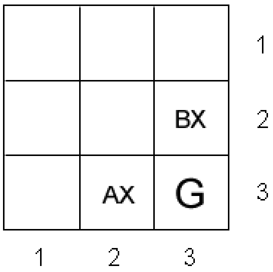

Figure 1: Generalization in the Grid world by means of sharing of elements across states.

Generalization experiments were designed in two phases (see Figure 2). In Phase 1 the agent is trained with an initial Grid-world layout, and in Phase 2 this initial Grid-world layout is changed. The *G* in the lower right corner of each Grid-world layout represents the goal state and is the same for all layouts, for all experiments and phases. Phase 1 has the same elements as Phase 2 for all locations, unless otherwise stated. All experiments aim to test whether Phase 2 will converge faster to the optimal policy through generalization with Phase 1. Two groups are employed for each experiment. In Group 1 there is supposed to be generalization from Phase 1 to Phase 2 due to an environment change, which leaves an aspect of the environment layout intact, but changes another. Group 2, on the other hand, changes the environment, but leaves no aspect of Phase 1 similar in Phase 2. If convergence is faster in Phase 2 of Group 1 than in Phase 2 of Group 2, it can be seen as an indicator of generalization.

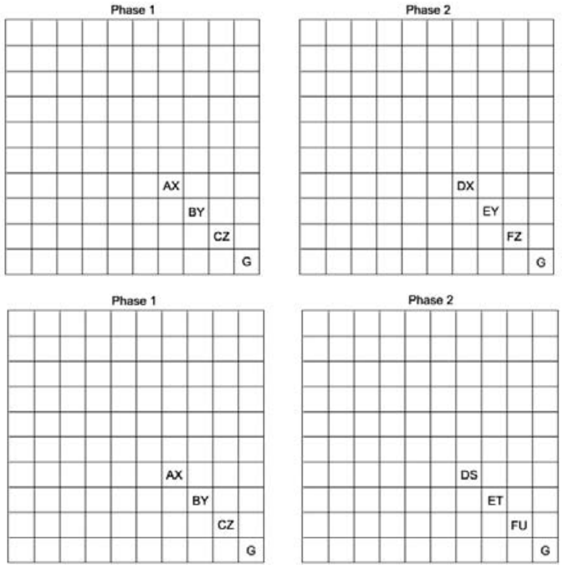

Figure 2: Generalization experiment in the 10×10 Grid-world, Group 1 top, Group 2 bottom.

To test generalization means to test whether IDQ finds the optimal policy faster in Phase 2 of Group 1 than in Group 2 (by the test statistic OPR). The independent variable is the difference in

Table 2 shows a sample mean OPR of 496.375 for Group 1 and 1059.75 for Group 2. The respective variances for Group 1 and Group 2 are 15107.69643, 134965.0714. It is assumed that the data is normally distributed for all eight samples in both groups. In order to check whether the heterogeneity of variance is significant at the .05 level, an F Max test is performed. In Table 2 the f value is reported to be 0.009841503. The degrees of freedom for the numerator is (n_1 – 1) = 7, and (n_2 – 1) = 7 for the denominator. According to the f distribution this gives a critical value of 3.79 at the .05 level of significance, which the f value does not exceed, so the heterogeneity of variance is not significant. A one-tailed t-test is therefore performed (it is expected that the difference between Group 2 and Group 1 is positive). Table 2 presents a t-value of 4.397315147. At (n_1 + n_2 – 2) = 14 degrees of freedom, this gives a critical value of 1.761 at the .05 level of significance. The t-value well exceeds the critical value at the .05 level, and also at the .01 level (critical value: 2.624), as well as at the .001 level (critical value: 3.787). It is therefore concluded that *IDQ generalizes.*

Table 2: IDQ in a 10×10 Grid-world (s. stands for sample and sq. for squares).

Sample	**OPR** **Group 1** **Phase2**	**OPR** **Group 2** **Phase 2**
1	455	1379
2	772	1787
3	564	768
4	389	715
5	470	1066
6	428	736
7	475	993
8	418	1034
$\bar{X}$	496.375	1059.75
S	122.91336	367.3759266
S^2	15107.693	134 965.0714
MAX	772	1787
MIN	389	715
sum of s. sq.	2 076 859	9 929 316
sq. of sum of s.	15 768 841	71 876 484
n	8	8
F Max test	0.009841503	
t	4.397315147	

REFERENCES

Sutton, R. S., and Barto, A. G., 1998. *Reinforcement Learning: An Introduction*. The MIT Press, Cambridge, MA.

Watkins, C. J. C. H., and Dayan, P., 1992. Q-learning. *Machine Learning*, *8*, 279-292.