Noisy reinforcements in Reinforcement Learning: some case studies based on Gridworlds

ÁLVARO MORENO, JOSÉ D. MARTÍN, EMILIO SORIA,
RAFAEL MAGDALENA, MARCELINO MARTÍNEZ.
Digital Signal Processing Group
Dept. of electronic Engineering
University of Valencia
C/ Dr. Moliner, 50. 46100 Burjassot
(Valencia) Spain

Abstract: Reinforcement Learning (RL) has as its main objective to maximize the rewards of an objective function. This is achieved by an agent which carries out a series of actions to modify the state of the environment. The reinforcements are the cornerstone of the RL. In this work, a modification of the classic scheme of RL is proposed. Our proposal is based on applying a reinforcement with uncertainty; namely, it adds a random signal to the reinforcement. This uncertainty makes the agent incapable to learn. In this work, we consider this variation in the reinforcements as noise added to the reinforcement; therefore, the proposal is to filter the reinforcement signal in order to remove the noise. In particular, a moving average filter is used; this is one of the most simple filters used in Digital Signal Processing. The proposed variation is tested in a classical RL problem, namely, Gridworlds with different characteristics regarding size, structure and noise level. Results show that our proposed approach finds the optimal solution under some conditions in which the classical procedure cannot find it.

Key–Words: Reinforcement Learning, Digital Signal Processing, Gridworld.

1 Introduction

Reinforcement Learning (RL) may be classified within the unsupervised learning, and it has its origins in psychology, namely in the "reinforced" theories of the trial-and-error learning of Edward Thorndike [5]. The reinforcement learning consists basically of two elements, the environment and the agent, as it is shown in the diagram of figure 1. Both elements are interconnected via perception and action. The part of perception provides the agent with information of the state (s_t) in which it is at time (t), and the reinforcement (r_t) is the reward or punishment received after a certain action (a_{t-1}) has been taken at the previous time. The actions (a) that the agent performs are the way to interact with the environment to obtain a certain target and the policy (π) is the strategy that follows the agent to obtain that target. The objective of all reinforcement algorithms is to maximize the reinforcements (rewards), and in particular, to maximize the so-called expected return R_t:

$$R_t = \sum_{k=0}^{T} \gamma^t \cdot r_{t+k+1} \qquad (1)$$

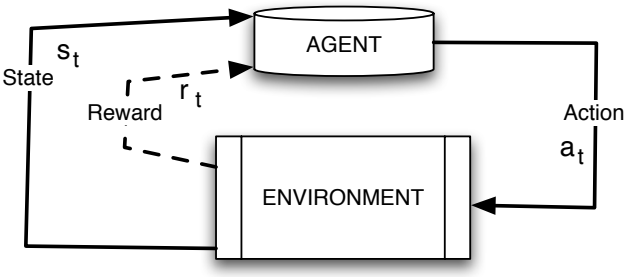

Figure 1: Basic diagram of a Reinforcement Learning system.

where T represents the final state, and $0 \leq \gamma \leq 1$ is a discount factor that measures the relevance of future actions. One of the most interesting and remarkable aspects of RL is that the algorithm is capable of programming the agent just using rewards and punishments without any explicit information about the way to achieve its objective.

Common applications of RL are, for instance, agents which play backgammon as a great master [6], or a mechanical biped that can learn to walk [4]. Other applications of RL involve the development of an op-

timal marketing policy [2, 1]. One of the main advantages of these systems is that they are able to learn by themselves or by a given experience. Thus, they can use the acquired knowledge and elaborate strategies to find an optimal policy.

Most of the experiments of RL that can be found in the literature are based on considering that the reinforcement has a constant value, but there are problems in which the reinforcement signal cannot be considered constant; in addition, the noise actually appears in the measures and it is absolutely necessary to diminish its effects for a successful performance of the system.

The proposal of this work is to filter the reinforcements as the way to find an optimal policy. As it is shown in Section 4, classical RL algorithms have convergence problems in the presence of noise. On the contrary, our proposed methodology can find the optimal policy by reducing the influence of noise in the reinforcements.

The remainder of the paper is outlined as follows. In Section 2, the proposed methodology is presented; Section 3 shows the experimental setup; in Section 4, the achieved results are presented, and we end up the paper with some conclusions in Section 5.

2 Proposed methodology

RL usually asumes that the reinforcements remain fixed throughout the learning. In this work, we take into account a temporary variation of these reinforcements. The reinforcement signal has been evaluated from the point of view of Digital Signal Processing, using the result of filtering the reinforcement signal corresponding to a certain state (it is necessary to remember that it assumes that exists a time variation in this reinforcement). The simplest filtering is the Moving Average (MA) and it will be used throughout the paper.

The expression of the simple average over the reinforcements is the following:

$$\overline{r}_t(s) = \frac{1}{N} \sum_{i=t-N+1}^{t} r_i \qquad (2)$$

being N the number of samples used in the average, t the time instant in which average is calculated and s the state which the reinforcement is associated to.

In Eq. 2, all the values have the same weight when they are processed. This feature is different from other filtering methods, such as the exponential average. The MA can be easily programmed and requires very few resources of the computer.

In this work, we focus on RL algorithms based on Temporal Differences. This kind of algorithms does not need a complete model of the environment to find an optimal policy, since they update the action-value function or the value function iteration by iteration [5]. In particular, the Q-learning algorithm is used here in its simple version (one step) [7].

3 Experimental Setup

In order to test the proposed approach, a classical RL problem is used: the Gridworld. There are many references about the use of RL in Gridworld environments [3, 5].

In this environment, the agent can move within this virtual space by taking only four actions: go up, go down, go right or go left (step by step). When an episode starts, the agent is in a certain initial state; in particular, we consider the upper-left corner as the initial state. All the reinforcements returned by the environment are null (the noise has not yet been taken into account) except in three cases:

1. The agent reaches the lower-right corner of the Gridworld, which is the desired final state (reinforcement equal to $+1$).

2. The agent leaves the state matrix (reinforcement equal to -1). In this case, the agent remains in the same state that it was before taking the action.

3. The agent tries to access forbidden or inaccessible states.

It is also possible to consider that the Gridworld has certain virtual walls which return a negative reinforcement when the agent collides with them. Figures 2 and 3 show diagrams of the used Gridworlds; although they represent a size 5×5 for the sake of simplicity, the previous description is also valid for any other size.

In this work, we consider two kinds of noise. First, a Gaussian noise is added to the reinforcement signal; this noise presents a zero mean and a variable standard deviation between 0 and 0.3; this maximum value of the standard devation is adequate because reinforcements are not completely masked by the noise, but the agent can have problems if the noise is not adequately processed. Second, an impulsive noise is also taken into account; the amplitude of this noise varies between 0.25 and 1, and the appearance probability is 20% and 10%, alternately.

The RL algorithm used in this work is the Q-learning (one step), being the action-value function

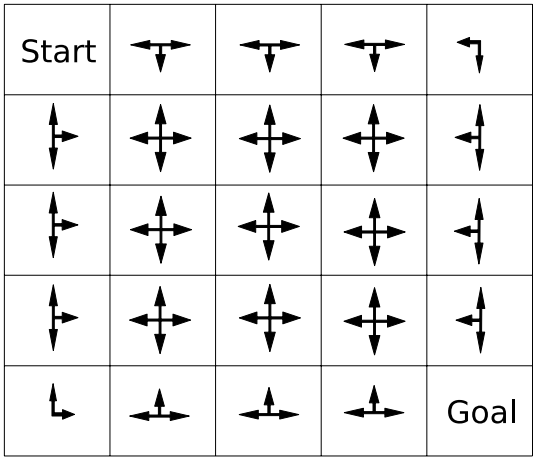

Figure 2: Classical Gridworld (size 5×5).

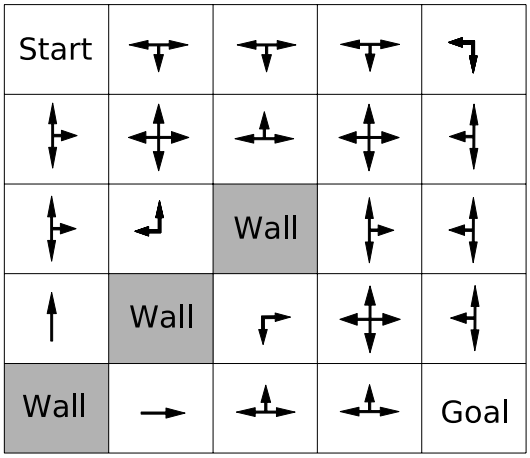

Figure 3: Gridworld (size 5×5) with walls.

updated as follows:

$$Q(s_t;a_t) \leftarrow Q(s_t;a_t) + \alpha[r_{t+1} + \gamma max_a Q(s_{t+1};a_{t+1}) - Q(s_t;a_t)] \quad (3)$$

where α is the learning rate, Q is the action-value function, s_t is the state at instant t, a_t is the action at instant t, and γ is the discount factor.

An $\epsilon - greedy$ policy is used to take the actions. It means that the choice of the action is based on analyzing which action provides the highest value of the action-value function the majority of the time (85% of the time in our case, which corresponds to $\epsilon = 0.15$), whereas random actions are taken the 15% of the time. These random actions are called exploratory actions and allow to find the long-term optimal solutions. This value of ϵ is optimum for our problem (values between 0 and 0,5 were tested).

In order to ensure the statistical rigor, measures were taken for 2500 episodes, being 10^6 the maximum number of iterations per episode. Moreover, in order to show the behavior of the algorithms under different conditions, different sizes of Gridworlds (with and without walls) were considered (5×5, 8×8 and 10×10).

Our analysis takes into account the average number of iterations used to complete the 2500 episodes (15 measures are carried out) for different noise parameters. It also analyzes whether the policy obtained by the agent is correct in all the measures (if it is not the case, the policy is labeled as wrong).

4 Results

4.1 Environment with Gaussian noise in the reinforcement signal

In this section, the classical Q-learning algorithm and our proposed algorithm (Q-learning with noise filtering) are compared when the environment is contaminated by Gaussian noise. Results are shown in Tables 1 and 2. In these two tables, as well as in the following ones, C means that 100% of the obtained policies were correct, W stands for some wrong policy, and S means that the algorithm was stagnated in a certain measure, and therefore, it did not achieve any solution.

Table 1: Results obtained by the classical Q-learning algorithm. The standard deviation of the Gaussian noise is indicated by σ.

			σ		
			0.1	0.2	0.3
No Wall	Gridworld Size.	$5 \cdot 5$	C	C	W
		$8 \cdot 8$	W	W	W
		$10 \cdot 10$	W	W	W
Wall		$5 \cdot 5$	C	W	W
		$8 \cdot 8$	W	W	W
		$10 \cdot 10$	W	W	W

C Correct policy. W Wrong policy. S Stagnation.

Table 2: Results obtained by the Q-learning algorithm filtering the reinforcement signal in order to remove the effect of the Gaussian noise.The standard deviation of the Gaussian noise is indicated by σ

			σ		
			0.1	0.2	0.3
No Wall	Gridworld Size.	$5 \cdot 5$	C	C	C
		$8 \cdot 8$	C	C	C
		$10 \cdot 10$	C	C	W
Wall		$5 \cdot 5$	C	C	C
		$8 \cdot 8$	C	C	C
		$10 \cdot 10$	C	C	S

The comparison of Tables 1 and 2 shows that our approach was clearly better, since it obtained a higher number of correct policies even with standard deviations considerably high ($\sigma = 0.3$ was a value relatively high compared with the reinforcement signal, whose maximum value was 1, which corresponded to the goal state). Moreover, it should be emphasized that there were not relevant differences between Gridworlds with or without walls.

4.2 Environment with impulsive noise in the reinforcement signal

In this case, the classical Q-learning algorithm and our approach are compared again, but when there is a presence of impulsive noise in the reinforcement signal. Tables 3 and 4 show the results achieved for different values of the amplitude of the pulse (strength of the pulse), and for different appearance probabilities (10% ratio or 20% ratio).

Table 3: Results obtained by the classical Q-learning algorithm in the presence of impulsive noise.

			Strength Of the Pulse.			
			0.25	0.50	0.75	1.00
10% Ratio.	No Wall	$5 \cdot 5$	C	W	W	W
		$8 \cdot 8$	W	W	W	W
		$10 \cdot 10$	W	W	W	W
	Wall	$5 \cdot 5$	W	C	W	W
		$8 \cdot 8$	W	W	W	W
		$10 \cdot 10$	W	W	W	W
20% Ratio.	No Wall	$5 \cdot 5$	C	W	W	W
		$8 \cdot 8$	W	W	W	W
		$10 \cdot 10$	W	W	W	W
	Wall	$5 \cdot 5$	C	W	W	W
		$8 \cdot 8$	W	W	W	W
		$10 \cdot 10$	W	W	W	W

Gridworld Size.

C Correct policy. W Wrong policy. S Stagnation.

Table 4: Results obtained by the Q-learning modified by the noise filtering.

			Strength Of the Pulse.			
			0.25	0.50	0.75	1.00
10% Ratio.	No Wall	$5 \cdot 5$	C	C	C	C
		$8 \cdot 8$	C	C	C	C
		$10 \cdot 10$	C	S	S	S
	Wall	$5 \cdot 5$	C	C	C	C
		$8 \cdot 8$	C	C	C	S
		$10 \cdot 10$	S	S	S	S
20% Ratio.	No Wall	$5 \cdot 5$	C	C	C	C
		$8 \cdot 8$	C	C	S	W
		$10 \cdot 10$	C	S	S	S
	Wall	$5 \cdot 5$	C	C	C	C
		$8 \cdot 8$	C	C	W	C
		$10 \cdot 10$	S	S	S	S

Gridworld Size.

Tables 3 and 4 show the usefulness of our approach even more than in the case of Gaussian noise. Although Table 4 shows that our approach was able to solve the problem in many cases in which the classical Q-learning could not find a correct policy, it is remarkable that our approach tended to stagnation when the size of the Gridworld was large (especially, 10×10). It might be due to two reasons: 1) the incapacity of the algorithm to eliminate the wrong knowledge acquired in the first episodes of the learning because of

the noise; and 2) the small variations in the filtered noise signals used by the RL system. It should be emphasized again that there were not relevant differences between Gridworlds with or without walls; it suggests the use of more complex walls in future Gridworlds in order to test the capabilities of our approach in a more complex scenario. The same can be applied to the appearance probabilities (higher ratios may show more differences, and hence, more interesting conclusions).

5 Conclusions

In this work, we have analyzed the addition of a noise signal to the reinforcement signal in RL systems. This noise stands for the uncertainty that may appear in some real problems.

Results have shown that the classical RL approach does not work in this scenario. We have proposed to filter the reinforcement signal as a simple way to eliminate the influence of the noise, thus solving the problem. In particular, one of the most simple filters (MA) has been used. Results have shown that this approach makes the RL system more robust, since it can find a correct policy in the majority of case studies.

This work is a first approach to solve RL problems in noisy environments. It is based on Digital Signal Processing. Future work will be devoted to the study of other kind of filters in order to remove the noise from the reinforcement signal.

Acknowledgements

This research has been partially supported by the project "NDPG: Neuro-Dynamic Programming Group: Aplicaciones prácticas de programación neuro-dinámica y aprendizaje reforzado en minería web y marketing" with reference number GVA05/009.

References:

[1] N. Abe, N. Verma, C. Apte, and R. Schroko. Cross channel optimized marketing by reinforcement learning. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 767–772, New York, NY, USA, March 2004. ACM Press.

[2] G. Gomez-Perez, J. D. Martín-Guerrero, E. Soria-Olivas, E. Balaguer-Ballester, A. Palomares, N. Casariego, and D. Paglialunga. An approach based on reinforcement learning and self-organizing maps to design a marketing campaign. In *2nd International Conference on Machine Intelligence ACIDCA-ICMI 2005*, pages 259–265, Tozeur, Tunisia, Nov 2005.

[3] L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.

[4] J. Marimoto, G. Cheng, C. G. Atkeson, and G. Zeglin. A simple reinforcement learning algorithm for biped walking. In *In Proc. 4th IEEE Int. Conf. on Robotics and Automation*, volume 3, pages 3030–3035, New Orleans, LA, USA, April 2004.

[5] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introducion*. M.I.T. Press, 1998.

[6] G. Tesauro. Td-gammon, a self-teaching backgammon program, achieves master-level play. *Neural Computation*, 6(2):215–219, March 1994.

[7] C. J. C. H. Watkins and P. Dayan. Technical note: Q learning. *Machine Learning*, 8:279–292, 1992.