Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 109C (2017) 1146–1151

Procedia Computer Science

www.elsevier.com/locate/procedia

International Workshop on Adaptive Technology (WAT 2017)

An Adaptive Implementation of ε-Greedy in Reinforcement Learning

Alexandre dos Santos Mignon[a,*], Ricardo Luis de Azevedo da Rocha[a]

[a]*Escola Politécnica da Universidade de São Paulo, São Paulo, Brasil*

Abstract

This paper presents a method called adaptive ε-greedy for better balancing between exploration and exploitation in reinforcement learning. This method is based on classic ε-greedy, which holds the value of ε statically. The solution proposed uses concepts and techniques of adaptive technology to allow controlling the value of ε during the execution. We present experiments numerically comparing the two methods for the n-armed bandit problem in a stationary environment. The adaptive ε-greedy method presents better performance as compared to the classic ε-greedy. For a nonstationary environment, we use an algorithm to detect the change point and adaptively modify the state of the agent to learn from the new rewards received. A recent related work was also evaluated in the paper.

Peer-review under responsibility of the Conference Program Chairs.

Keywords: Reinforcement Learning; ε-Greedy; N-armed Bandit; Adaptive Technology.

1. Introduction

Reinforcement learning is a form of machine learning in which an agent learns from its interaction with an environment to achieve a given goal[1]. By interacting with the environment through actions taken, the agent receives numerical rewards. The agent's goal is to maximize the total number of rewards received.

A distinctive challenge that arises in reinforcement learning lies in balancing between exploration and exploitation[1]. Exploitation means that the agent selects the action that has the highest average rewards. Exploration means that the agent selects an action randomly, regardless of the reward values obtained previously. Exploitation is the best alternative to undertaking to receive a good reward right away. However, to find out which actions are best, one needs to run the exploration mode. In this mode, one can find better action and better results in the long run. Because it is not possible to run exploration and exploitation mode at the same time, one must decide which mode to use to perform an action at a given time.

* Corresponding author. Tel.: +55-11-3091-9084; fax: +55-11-3091-5713.
E-mail address: amignon@usp.br

Peer-review under responsibility of the Conference Program Chairs.
10.1016/j.procs.2017.05.431

A simple method for performing the balancing between exploration and exploitation is the method called ε-greedy[1]. This method behaves most of the time greedily, yet now and then, with a small probability ε, randomly selects one action among all actions. However, the value of this probability ε is static.

Adaptive technology deals with techniques and devices that allow a system to modify its behavior, in response to some input stimulus or its operating history, without any external interference[2]. Adaptive technology allows a system with static rules to become a system with dynamic rules.

This paper introduces the adaptive ε-greedy method. This method is based on the classic ε-greedy method but allows the value of ε to be changed according to the rewards received from the environment. We numerically compared the two approaches for the n-armed bandit problem. The adaptive ε-greedy presented superior performance to the classic ε-greedy in a stationary environment. For a nonstationary environment, we use an algorithm to detect the change point[3] and adaptively modify the state of the agent to learn from the new rewards received.

The paper is organized as follows. Section 2 presents the adaptive ε-greedy method and algorithm. Section 3 presents the method used in the experiments. Section 4 presents the results of the experiments. Section 5 discusses related work. Finally, Section 6 presents the conclusions and future work.

2. Adaptive ε-Greedy Method

Using the concepts and techniques of adaptive technology[2,4], we developed the method called adaptive ε-greedy. This method is based on the classic ε-greedy but allows the value of ε to vary in a controlled way throughout the execution. Changing the value of ε is accomplished by performing an adaptive action triggered throughout the process. A new value for ε is calculated using the rewards received from the environment.

The adaptive ε-greedy method presented here is a modified and improved version of the method previously presented[5], which dealt only with stationary environments. It has two configurable parameters: l and f. We added to the classic ε-greedy exploration mode one adaptive action that can change the value of ε. Parameter l is used to set how many times to run the exploration mode before performing the adaptive action that changes the value of ε. Parameter f is used to regularize the calculated values of the rewards received to obtain an adequate value of the function that produces a new value for ε.

Algorithm 1 Adaptive ε-greedy

```
1: max_prev ← 0
2: k ← 0
3: if normal distribution ≤ ε then
4:     max_curr ← Q_t(A*_t)
5:     k ← k + 1
6:     if k = l then
7:         Δ ← (max_curr − max_prev) ∗ f
8:         if Δ > 0 then
9:             ε ← sigmoid(Δ)
10:        else
11:            if Δ < 0 then
12:                ε ← 0.5
13:            end if
14:        end if
15:        max_prev ← max_curr
16:        k ← 0
17:    end if
18:    randomly selects an action
19: else
20:    selects A*_t
21: end if
```

Algorithm 1 presents the algorithm used in the adaptive ε-greedy method. This algorithm is used to choose an action a when it is in a state s. It decides whether to perform exploitation or exploration mode. If it decides for the exploitation mode, it selects the action with highest average reward (A_t^*). If it decides for the exploration mode, it selects an action randomly. When the algorithm is in the exploration mode, it can perform an adaptive action, depending on its configuration, which modifies the value of ε.

The variables max_{prev} and k are static, being used to store the state of the algorithm. Variable k is used to count how many times the algorithm performs the exploration mode, after the last time it changed the value of ε.

When the k variable reaches the limit specified in the l parameter, the algorithm decides whether to alter the value of ε. To do so, the difference (Δ) between the highest average rewards (max_{curr}) and the highest previous rewards average (max_{prev}), obtained in the last time the Value of the variable k has reached the limit l. This difference is multiplied by the value of the parameter f to regularize its value. If the Δ value is greater than zero, a new value is calculated for ε.

If the Δ value is less than zero, it means that after the last setting of a new value for ε, the algorithm has more often selected actions that are not optimal, and we must thus continue to search for optimal actions. For this, $\varepsilon = 0.5$ is

defined. If the Δ value equals zero, its current value remains. After setting the new value of ε, variable k is zeroed and the variable max_{prev} receives the value of the variable max_{curr}.

A new value for ε is defined by using a sigmoid function: $sigmoid(x) = \frac{1.0}{1.0 + \exp(-2 * x)} - 0.5$. According to this function, the value of ε can vary between 0.0 and 0.5. Empirically, it has been found that there is no significant gain if the value of ε is greater than 0.5.

3. Method of Experiments

This section presents the methods used in the experiments. Two types of experiments were performed: one for a stationary environment and the other for a nonstationary environment. The purpose of the experiments is to numerically compare the classic ε-greedy and adaptive ε-greedy methods for the n-armed bandit problem.

In both experiments, the configurable parameters of the adaptive ε-greedy method were defined as $l = 10$ and $f = 7$. These values were empirically stipulated as better after experiments performed between a range of values. For the value of parameter l, we tested values in the range between 1 and 20. For the value of parameter f, we tested values in the range between 1 and 25.

The experiments were performed with a set of 2000 randomly generated tasks for the n-armed bandit problem with $n = 10$. For each action, the values are selected according to a normal (Gaussian) distribution with mean 0 and variance 1. Calculating the average rewards received for each action, one can trace the performance and behavior of the methods. The performance and the behavior of the methods were verified by executing 1000 plays. Data were obtained by the median of the 100-fold run of the experiments.

3.1. Stationary Environment

In the experiment performed in a stationary environment, two classical ε-greedy methods ($\varepsilon = 0.1$ and $\varepsilon = 0.5$) were compared with the adaptive ε-greedy method ($\varepsilon = 0.5$). The value of $\varepsilon = 0.5$ in the adaptive method is just an initial value for ε, as it is modified throughout the execution. Initial values for ε in the range of 0.1 to 0.9 were tested, and the results were similar. We adopted the starting value of 0.5 because it is the largest value returned by the sigmoid function used to define a value for ε. The methods calculate the actions values estimate using the sample average technique [1].

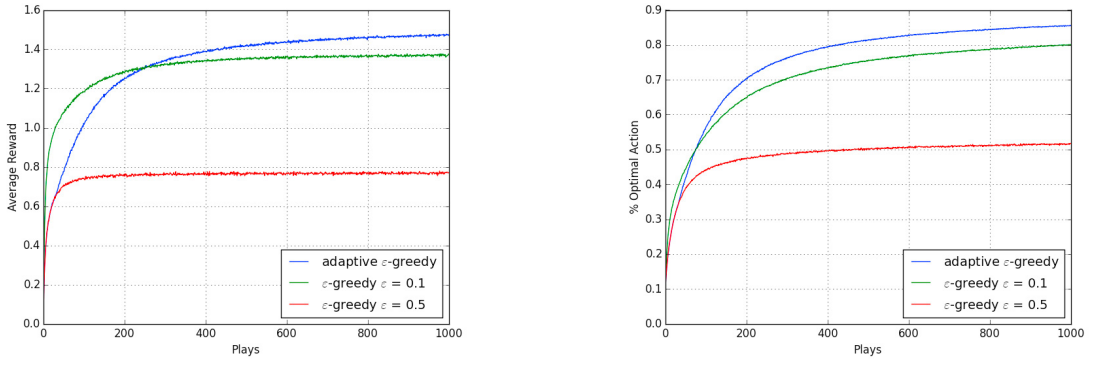
3.2. Nonstationary Environment

The sample average technique is appropriate to a stationary environment, but not if the environment changes over time. In such cases, we use the technique of assigning a greater weight to the more recent rewards. One of the most popular ways of assigning this weight is to use a constant step-size parameter (α) [1].

However, in a nonstationary environment, it is important to detect any change. There are three types of possible environment changes [3]: 1) the estimated reward of the best option so far decays abruptly, and another option becomes better; 2) the estimated reward of the best option so far is modified, but the best option remains the best; 3) The estimated reward of the best option so far does not change, but the reward of some other option increases to the point that the other option becomes the best. Only the first type of change is considered herein.

The change point the environment is detected by using the algorithm presented by Hartland et al. [3]. This algorithm uses the Page-Hinkley (PH) statistical test [6]. The PH test involves two parameters. Parameter λ controls the trade-off between exploration and exploitation. Parameter δ is meant to make the PH-test more robust when dealing with slowly varying environments.

In the experiment performed in the nonstationary environment, we compared the ε-greedy ($\varepsilon = 0.1$) method with the adaptive ε-greedy method, both using the PH test to detect the change point. The strategy adopted is to calculate the value of the reward estimates using the sample average technique until a change point is detected. When a change point is detected, an adaptive action is performed. In this adaptive action, the values of the reward estimates are assigned zero, and a new calculation using sample average is started. In the adaptive ε-greedy method, the value of the variables max_{prev} and k is assigned zero, and the value of ε is assigned 0.5, that is, the initial configuration of the algorithm is restored.



(a) (b)

Fig. 1: (a) Average performance, in a stationary environment, of ε-greedy and adaptive ε-greedy methods to expected reward; (b) Average performance, in a stationary environment, of ε-greedy and adaptive ε-greedy methods in selecting optimal actions.

We have determined that the environment change after 500 plays. For the PH test, we set the value of λ to 35, because it obtained a better result in a range between 5 to 80; and the value of δ to 0.05, because it obtained a better result in a range between 0 to 0.5.

4. Results

This section presents the results of the experiments described in the previous section. We implemented the algorithm presented in Section 2 in the Python 2.7 programming language and ran the experiments on a machine with Intel Core i5 2.30 GHz, with 4 GB of RAM, running Windows 10.

4.1. Stationary Environment

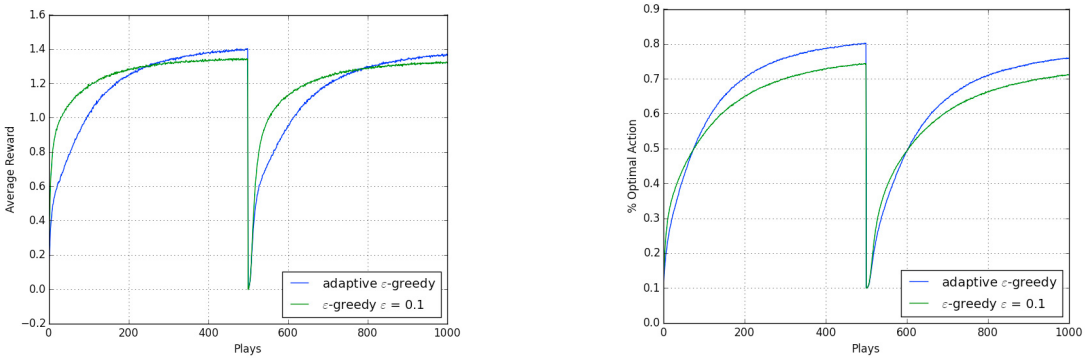
Fig. 1a shows the expected reward gradient with experience. The adaptive ε-greedy method presented superior performance to the other two methods. It can be seen that it took more than 200 plays for the adaptive ε-greedy method to outperform the classical ε-greedy method with $\varepsilon = 0.1$. In the end, the adaptive ε-greedy method reached a reward of approximately 1.47, while the classical ε-greedy method with $\varepsilon = 0.1$ eventually obtained a reward of approximately 1.37. The classical ε-greedy method with $\varepsilon = 0.5$ eventually obtained a reward of only 0.77, approximately.

Fig. 1b shows that the adaptive ε-greedy method finds optimal actions more quickly and ultimately reaches an approximate 85% selection of optimal actions. Meanwhile, the best-performing classic ε-greedy method ($\varepsilon = 0.1$) reached a percentage of approximately 80% optimal action selection. It can also be observed that throughout the process, the adaptive ε-greedy method selects optimal actions a greater number of times, making the difference in choosing optimal actions greater than 5 percentage points.

4.2. Nonstationary Environment

Fig. 2a shows the expected reward gradient with experience. After the configuration change at step 500, the adaptive ε-greedy method reached a reward of about 1.37, exhibiting superior performance to the ε-greedy method with $\varepsilon = 0.1$, which attained a reward of about 1.32.

Fig. 2b shows that the adaptive ε-greedy method finds optimal actions sooner and, in the end, reached a percentage of about 76% optimal action selection. In the meantime, the ε-greedy method with $\varepsilon = 0.1$ eventually reached a percentage of about 71% optimal action selection.



(a) (b)

Fig. 2: (a) Average performance, in a nonstationary environment, of ε-greedy and adaptive ε-greedy methods with PH-test to the expected reward; (b) Average performance, in a nonstationary environment, of ε-greedy and adaptive ε-greedy methods with PH-test in selecting optimal actions.

5. Related Work

Although the ε-greedy method is often used for balancing between exploration and exploitation, the literature lacks of methods that adapt the exploration rate based on the learning progress[7]. Only a few methods such as ε-first or decreasing-ε consider the time to reduce the probability of exploration.

Tokic[7] presents a method called "Value-Difference Based Exploration" (VDBE). This method adapts the ε-greedy exploration parameter based on the temporal-difference error observed from the value functions backups. In this work, a new value for parameter ε is obtained after each learning step using equations based on the Boltzmann distribution of the value function estimates. At the beginning of the learning process, the value of the ε parameter is initialized to 1.0.

We perform experiments to numerically compare the adaptive ε-greedy and VDBE methods for the n-armed bandit problem in a stationary environment. The adaptive ε-greedy method was established according to the method presented in Section 3. For the VDBE method, two different settings were used for the α parameter: $\alpha = 0.33$ and $\alpha = 0.04$; both with the $\delta = 0.1$ parameter. The settings were chosen based on the best results presented by Tokic[7]. Data were obtained by the median of the 100-fold run of the experiments.

Fig. 3a shows the expected reward gradient with experience. The VDBE method with $\alpha = 0.33$ was the one that presented a superior performance at the beginning of the process, but at the end of the process, all the methods presented similar performance. Table 1a details the graph of Fig. 3a showing the average rewards and the highest reward obtained along the process and also the reward value gained in the last play. Although in the last play the methods perform similarly, the VDBE method with $\alpha = 0.33$ has the highest average reward seeing that it obtains greater rewards at the beginning of the process.

Fig. 3b shows that the VDBE method with $\alpha = 0.33$ finds optimal actions more quickly, but then stabilizes at a lower value than the other two methods. The VDBE method with $\alpha = 0.04$ finds optimal actions later, but at the end of the process, it performs better than the other two methods. The adaptive ε-greedy method finds optimal actions faster than the VDBE method with $\alpha = 0.04$, but at the end of the process, it marginally selects fewer optimal actions.

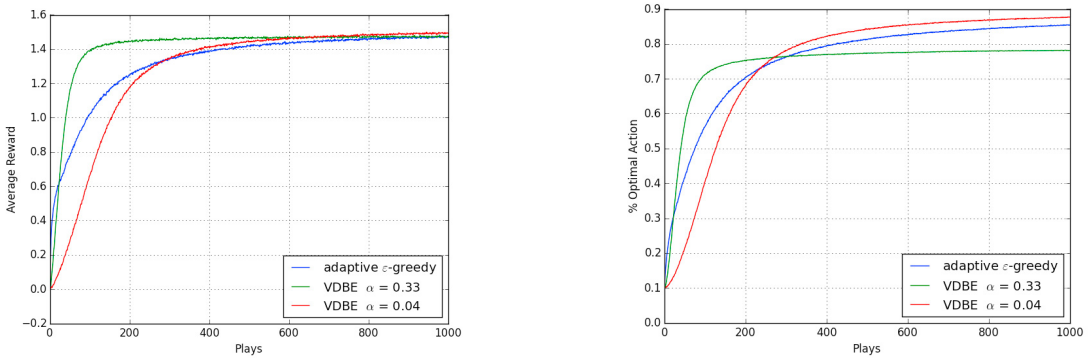
Table 1: (a) Data of the experiments of comparison of the adaptive ε-greedy and VDBE methods for the expected reward; (b) Data of the experiments of comparison of the adaptive ε-greedy and VDBE methods in selecting optimal actions.

Method	AVG	Highest	Last Play
adaptive ε-greedy	1.322	1.477	1.468
VDBE ($\alpha = 0.33$)	1.411	1.483	1.473
VDBE ($\alpha = 0.04$)	1.271	1.500	1.492

(a)

Method	AVG	Highest	Last Play
adaptive ε-greedy	75.5%	85.5%	85.4%
VDBE ($\alpha = 0.33$)	74.2%	78.2%	78.1%
VDBE ($\alpha = 0.04$)	74.8%	87.7%	87.7%

(b)



(a) (b)

Fig. 3: (a) Average performance, in a stationary environment, of adaptive ε-greedy and VDBE methods to expected reward; (b) Average performance, in a stationary environment, of adaptive ε-greedy and VDBE methods in selecting optimal actions.

Table 1b details the graph of Fig. 3b showing data from the experiments performed on the average performance of the methods in choosing the optimal action. The *AVG* column shows the average percentage obtained. The *Highest* column shows the highest proportion value obtained throughout the process. The *Last Play* column displays the percentage value gained in the last play.

Although the adaptive ε-greedy method selects optimal actions approximately two percentage points below that of the VDBE method with $\alpha = 0.04$, the average over the course of the process is similar. However, the average rewards obtained by the adaptive ε-greedy over the process is greater than that of the VDBE method with $\alpha = 0.04$.

6. Conclusion

A distinctive challenge of the area of reinforcement learning is the balancing between exploration and exploitation. ε-greedy is a simple method for this balancing; however, the ε value is static. In this work, we introduce the adaptive ε-greedy method. It allows the value of ε to be dynamic, adapting to the behavior of the environment.

The experiments performed showed that, in a stationary environment, the adaptive ε-greedy method performed better than the ε-greedy method. The adaptive version finds optimal actions in fewer plays and also selects optimal actions a greater percentage of times, with performance around 5 percentage points more in the last play.

In a nonstationary environment, we use the PH-test to detect the change point of the environment. After changing the environment, the adaptive version sooner matches an action close to the optimum and selects the optimal action a higher percentage of times, with a performance by 5% more on the last play.

As future work, we intend to study new adaptive learning policies to improve the performance of the adaptive ε-greedy method in a nonstationary environment and how to deal with types of change 2 and 3 presented in Section 5.

References

1. Sutton, R.S., Barto, A.G.. *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press; 1 ed.; 1998.
2. Neto, J.J.. A Small Survey of the Evolution of Adaptivity and Adaptive Technology. *Revista IEEE América Latina* 2007;**5**(7):496–505. (in Portuguese).
3. Hartland, C., Baskiotis, N., Gelly, S., Sebag, M., Teytaud, O.. Change Point Detection and Meta-Bandits for Online Learning in Dynamic Environments. *CAp* 2007;:237–250.
4. Luz, J.C.. *Adaptive Technology Applied to Compiler Code Optimization*. Master's thesis; Escola Politécnica, USP; 2004. (in Portuguese).
5. Mignon, A.S., Rocha, R.L.A.. ε-Greedy Adaptativo. In: *Memórias do VIII Workshop de Tecnologia Adaptativa - WTA 2014*. São Paulo, Brazil; 2014, p. 57–62. (in Portuguese).
6. Page, E.S.. Continuous Inspection Schemes. *Biometrika* 1954;**41**(1/2):100–115.
7. Tokic, M.. Adaptive ε-greedy Exploration in Reinforcement Learning Based on Value Differences. In: *Proceedings of the 33rd Annual German Conference on Advances in Artificial Intelligence*; KI'10. Berlin, Heidelberg: Springer-Verlag; 2010, p. 203–210.