Acta Technica *62*, No. 2B/2017, 639–656

Evaluation of multi armed bandit algorithms and empirical algorithm

Zhang Hong[2,3], Cao Xiushan[1], Pu Qiumei[1,4]

Abstract. Multi armed bandit (MAB) model is a model of reinforcement learning, which is proposed in the experimental sequence design of statistical learning theory. The goal of the model is to maximize the reward through a series of experimental sequences. The final question of the model is the trade-off between "exploitation" and "exploration". Based on the theory of Multi Armed Bandit model, this paper analyzes the commonly used algorithms of ε-greedy, UCB1, Softmax and Pursuit algorithm, and research its convergence under different parameters. The experimental result shows that the convergence rate and the regret degree of the ε-greedy algorithm are better than other ones in both static and dynamic situations.

Key words. Multi armed bandit, reinforcement learning, ε-greedy, Softmax, Pursuit..

1. Introduction

Multi armed bandit model is extensively used to the exploration and exploitation trade-off. It can help the agent explore new knowledge and exploit it to improve the future decision. It have many applications, for example product recommendation, online advertisement, smart grids, internet routing and so on.

In the statistical learning theory, the multi armed bandit (MAB) model, referred to as the bandits problem. It is about a problem that a gambler (or a few) at a row of slot machines decide which machine to play, how many times of each slot machine arms should pull and the order of pulling the arm . When the arm is pulled down, each slot machine will provide a random reward with a distributed at a certain probability, the goal of gambler is getting the maximum reward through a series of pulls order. Each time, the gambler will face an important trade-off

[1]School of information Engineering,Minzu University of China, 100081, China.
[2]Prof., School of information Engineering, Systems Science, Zhang Hong, China
B.S., School of information Engineering, Systems Science, Cao Xiushan, China
Dr., School of information Engineering, Intelligent System, Pu Qiumei, China
[3]E-mail: zhang_hong__@163.com
[4]E-mail: puqiumei@muc.edu.cn

between the "exploitation" of the machine with the highest expected reward and the "exploration" of obtaining more information about the expected benefits of other machines.

The initial model of the multi-arm slot machine was formed from the convergence of the overall selection strategy, which was constructed by Herbert Robbins in the literature "Some aspects of the sequential design of experiments" in 1952, and the optimal policy for maximizing the expected discounted reward (Gittins index) proposed by John C. Gittins. The model is further developed in the Markov decision processes (MDP). From then on all kinds of multi armed bandit algorithms have been proposed by scholars and some algorithms regret bound are well understood. However there are only a few of these algorithms experiment and evaluation. So in this paper, we will evaluate these algorithms from a comprehensive. At the first part, a brief of ε-greedy, UCB1, Softmax and Pursuit algorithm's theory were introduced. In the experiment part, each individual strategy is experimented by changing the algorithm's own parameters and observing how each parameter affects the performance of the algorithm. And then, these algorithms are compared by changing the number of arms and reward distributions, and a comprehensive evaluation is made in this section.

2. The multi armed bandit model theory

2.1. Reinforcing learning theory

Reinforcing Learning [1] is an unsupervised learning that takes input from changing environmental feedback, it is usually used to evaluate each step strategy by rewarding and punishing mechanisms. It maximizes the final returns by strategies that continually selecting for positive feedback, so that the problem is optimal solution. The objective of reinforcing learning is to enable the agent to take action at any time to maximize the cumulative reward. The control strategy is expressed as follow $\pi : S \to A$, according to the control strategy, when the agent at a certain point time observes that the environmental state is S_t, will take action a_t. After the action is performed, the reward value of the agent is $r_t = fr(s_t, a_t)$, the system state becomes $S_{t+1} = fs(s_t, a_t)$.

2.2. Multi armed bandit model theory

The multi armed bandit model [2], also known as the k-armed bandit, resembles the traditional slot machine (single-arm slot machine), but it generally has more than one lever. When pulled, each lever provides a reward based on the probability distribution associated with the particular lever [3]. Initially, the gambler did not have the priori knowledge about each levers.

The multi armed bandit model can be seen as a set of real distributions $B = \{R_1, \ldots, R_k\}$, and each distribution is associated with the rewards delivered by one of the k ($k \in N^+$) levers. $\mu_1, \ldots, \mu_k$ to be separately defined for each mean value of the reward distribution. Gambler repeatedly play a lever and observe the corresponding

reward. His goal is to maximize their reward income. The horizon H is the number of rounds that remain to be played. The model problem is formally equivalent to a state Markov decision process. R is the degree of regret [4]

$$R_T = T\mu^* - \sum_{t=1}^{T} \hat{r}_t \,. \tag{1}$$

It is defined as the expected difference between the sum of the optimal policy returns after T rounds pull down and the sum of the actual gains, $\mu^* = \max_k \{\mu_k\}$ represents the maximum expected value of all the arms, $\hat{r}_t$ indicates the reward after t experiment, T indicates the total number of trials.

At the same time, zero-regret strategy [5] refers to the strategy that the probability of each round the R/T tends to zero with probability 1 when the number of rounds of the experiment tends to infinity. Intuitively, if there are enough rounds, the zero regret strategy is guaranteed to converge to (not necessarily the only) optimal.

3. Related algorithms to multi armed bandit model

In this section, we will briefly introduce several well-known bandit algorithms that we have experimented in this work. These are ε-greedy strategy, ε_{n}-greedy strategy, UCB1 strategy, Softmax strategy and Pursuit strategy respectively.

3.1. ε-greedy strategy

ε-greedy is the most simple strategy that is widely used and proves to be very effective one. This algorithm compromises the exploration and exploitation with a probability of ε. At each trial the arm that currently offers the highest mean reward be selected (if there are more than one, then randomly selected one) with probability of $1-\varepsilon$, and selects a random arm with probability ε. It means that to exploit with the probability $1-\varepsilon$ and to explore with the probability of ε. The process can be represented as

$$p_i(t+1) = \begin{cases} p_i(t) & \text{for } i = \operatorname{argmax}_{j=1,\ldots,K} \widehat{\mu}_j(t) \\ \varepsilon/k & \text{otherwise} \end{cases} . \tag{2}$$

Here, the value ε is fixed. Actually, there are many different versions of ε-greedy strategies, for example ε_{n}-greedy [6], the ε_{n} value in the algorithm will change with the number of trials. At initialization, defined ε_{n} sequence is $i = 1, 2, \cdots n$, where $\varepsilon_{\mathrm{n}} = \min\{1, \frac{cK}{d^2 n}\}$, $c > 0$, $0 < d < 1$.

3.2. UCB1 strategy

The upper confidence bound (UCB) family of algorithms is proposed by Auer, Cesa-Bianchi & Fisher (2002) [7], which is used to solve uncertain problems in optimization problems. In this paper, the simplest algorithm is given and is called UCB1. When the algorithm is executed, the times of pull and the average value of

reward for each arm are recorded. The algorithm initially will pull each arm once. In the $t = k + 1$ subsequent pull, the arm $j(t)$ is selected to follow the following formula

$$j(t) = \arg \max_{i=1,\ldots k} \left(\widehat{\mu}_i + \sqrt{\left(\frac{2 \ln t}{n_i} \right)} \right), \tag{3}$$

where $\widehat{\mu}_i$ is the mean reward of each lever i, n_i is the number of lever i has been selected. Auer, Cesa-Bianchi & Fisher (2002) [8] proved that after T rounds pulled, the confidence of the confidence community algorithm is bounded, and the bounded is

$$\left[8 \sum_{i:\mu_i < \mu *} \left(\frac{In\, \mathrm{n}}{\Delta_i} \right) \right] + \left(1 + \frac{\pi^2}{3} \right) \left(\sum_{j=1}^{K} \Delta_i \right), \tag{4}$$

where $\Delta_i = \mu^* - \widehat{\mu}_i$.

3.3. Softmax strategy

Softmax algorithm [9] compromises the exploration and exploitation with the current lever mean reward. Each arm is selected with a probability proportional to its mean reward. If the mean reward of each arm is equal, the probability of selecting each arm is also equivalence and if the mean reward of some arms are obviously higher than other arms, the probability of their selection are also significantly higher. Softmax uses the Boltzmann distribution to select an arm, the probability update formula is as follows

$$p_i(t+1) = \frac{e^{\widehat{\mu}_i(t)/\tau}}{\sum_{j=1}^{k} e^{\widehat{\mu}_i(t)/\tau}}, \, (i = 1, \ldots n), \tag{5}$$

where $\widehat{\mu}_i(t)$ is the mean reward for the current arm and $\tau > 0$ is called a temperature parameter, controlling the randomness of the choice. When τ tends to zero, Softmax will tend to "exploitation". When it tends to infinity, Softmax will tend to " exploration".

3.4. Pursuit strategy

Pursuit algorithm[10] pursues a lowest regret degree by maintaining an explicit policy over the arms, in which the best arm updates though experience. At each time t, it updates the probabilities as:

$$p_i(t+1) = \begin{cases} p_i(t+1) + \beta(1 - p_i(t)) & \text{if} \quad i = \mathrm{argmax}_{j=1,\ldots K} \widehat{\mu}_j(t), \\ p_i(t+1) + \beta(0 - p_i(t)) & \text{otherwise.} \end{cases}$$

where $\beta \in 0, 1$ is the learning rate.

4. Experiment simulation and research

The experiment consists of two parts. In the first part, each individual strategy is experimented by changing the algorithm's own parameters and observing how each parameter affects the performance of the algorithm. In the second part, these algorithms are compared by changing the number of arms and reward distributions, and a comprehensive evaluation is made in this section.

4.1. Experiment setup

The reward distribution of all the arms is randomly generated and distributed as a Gaussian distribution. The mean reward of each arm is randomly generated from (0,1). The number of trails is set to 1000, and repeat this procedure for 1000 times, the number of arms, we set to 5, 10 and 20. As the multi armed bandit model come into practice, there will be a lot of static and dynamic differences, static conditions such as the router's channel selection problem, the number of routers is an constant, so the number of channels will be constant, the appropriate channel selection will make the information transform faster; The dynamic situation such as the stock selection strategy, due to the listed company will increasing as the time goes, so the stock number will simultaneously increasing. Reasonable stock combination of old and new stocks options will make investors gain more benefit. So the static and dynamic differences are set. In the static case, the number of gambling machines will remain the same, and the number of gambling machines will increase after a certain number of trials.

As Figs. 1, 2 and Figs. 3, 4 show, the ε-greedy, Softmax, Pursuit, ε_{n}-greedy, these four algorithms were simulated, the changing is the algorithm's own parameters (since the UCB1 algorithm has no parameters, so this simulation did not carry out), and then set the same number of arms, variance and the number of rounds, and the arm of k is 10, and the variance is 0.1 and 1, the number of rounds for 1000 times. The only difference between Figs. 1, 2 and Figs. 3, 4 is that there is a different variance, and it is clear that the regrets in Figs. 3, 4 are higher than those in Figs. 1, 2. In both graphs, it is noted that the results are consistent with each other, as its own parameter value decreases, performance tends to improve.

As Figs. 5–7 show, the ε-greedy, ε_{n}-greedy, UCB1, Softmax and Pursuit algorithms are simulated for the same number of arms and different variance under their best parameters. Set the k to 5, then the variance is 0.001, 0.01, 0.1, 0.2, 0.5 and 0.8 respectively, and the number of rounds is 1000 times. The results shows that, in the case of different variance, the ε-greedy algorithm is always the first to converge, and the regret is relatively low, while the UCB1 algorithm is slower and the regression is unstable.

Consider another case, gamblers have learned 10 machine reward distribution and the corresponding regret through their own trails. This time the casino has increased 10 new machines and the 10 slot machine incentive distribution is unknown. From now on the gambler will face the choice of 10 old machines and 10 new machines, he will face a trade-off between existing knowledge and access to new knowledge. This

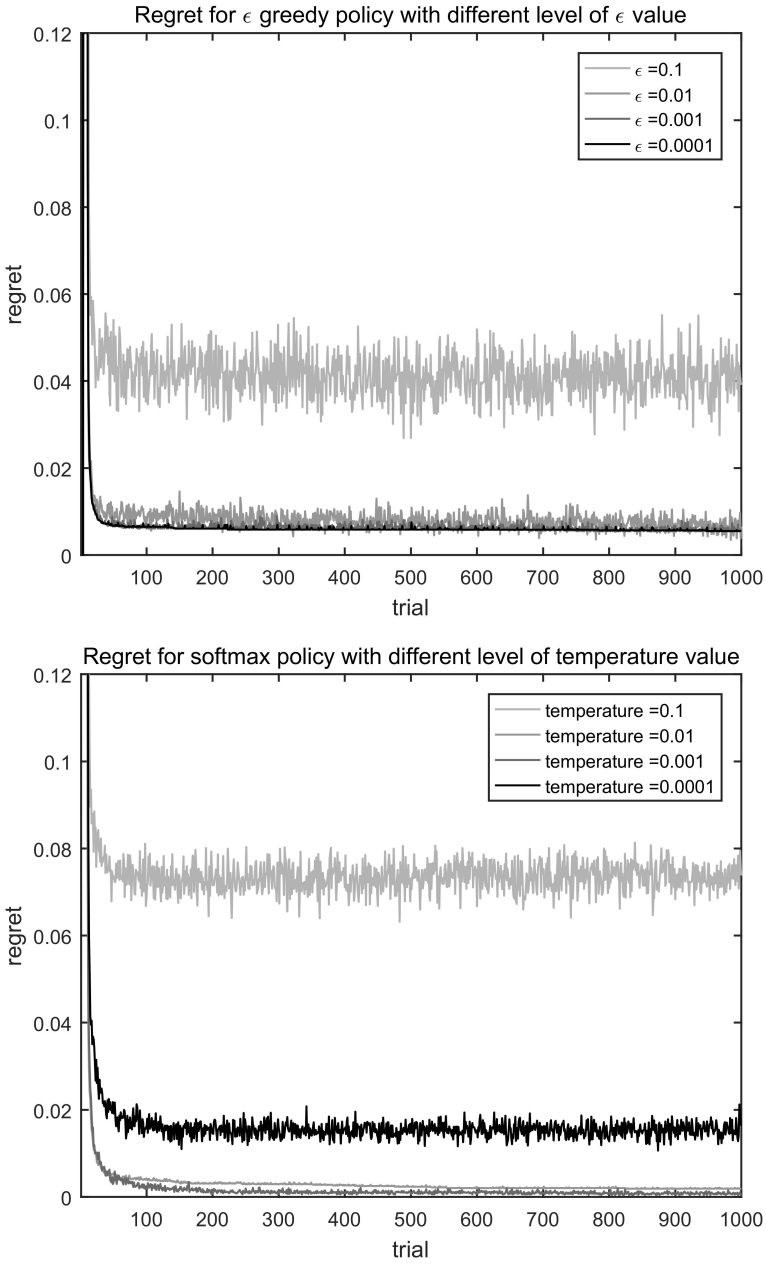

Fig. 1. Performance of each algorithm with different parameters(k=10,σ^2=0.1): top–ε-greedy, bottom–Softmax

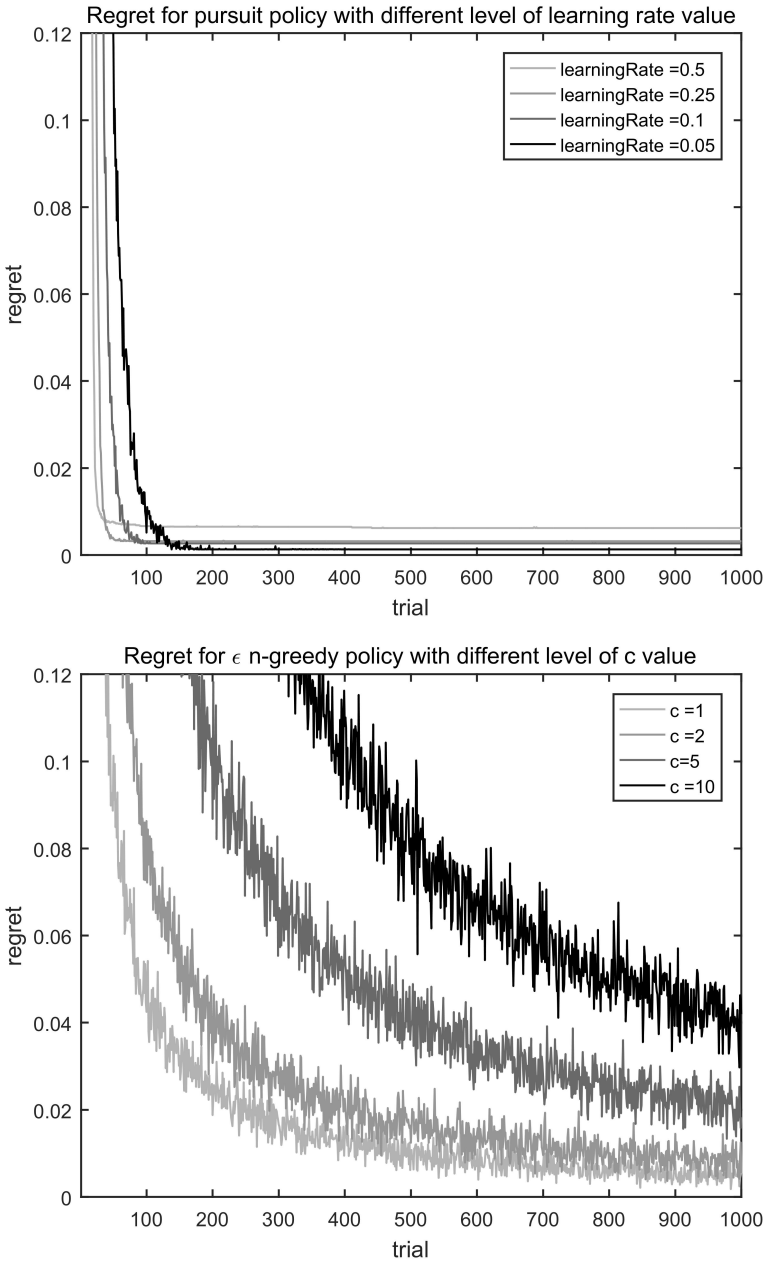

Fig. 2. Performance of each algorithm with different parameters(k=10,σ^2=0.1): top–pursuit, bottom–ε_{n}-greedy

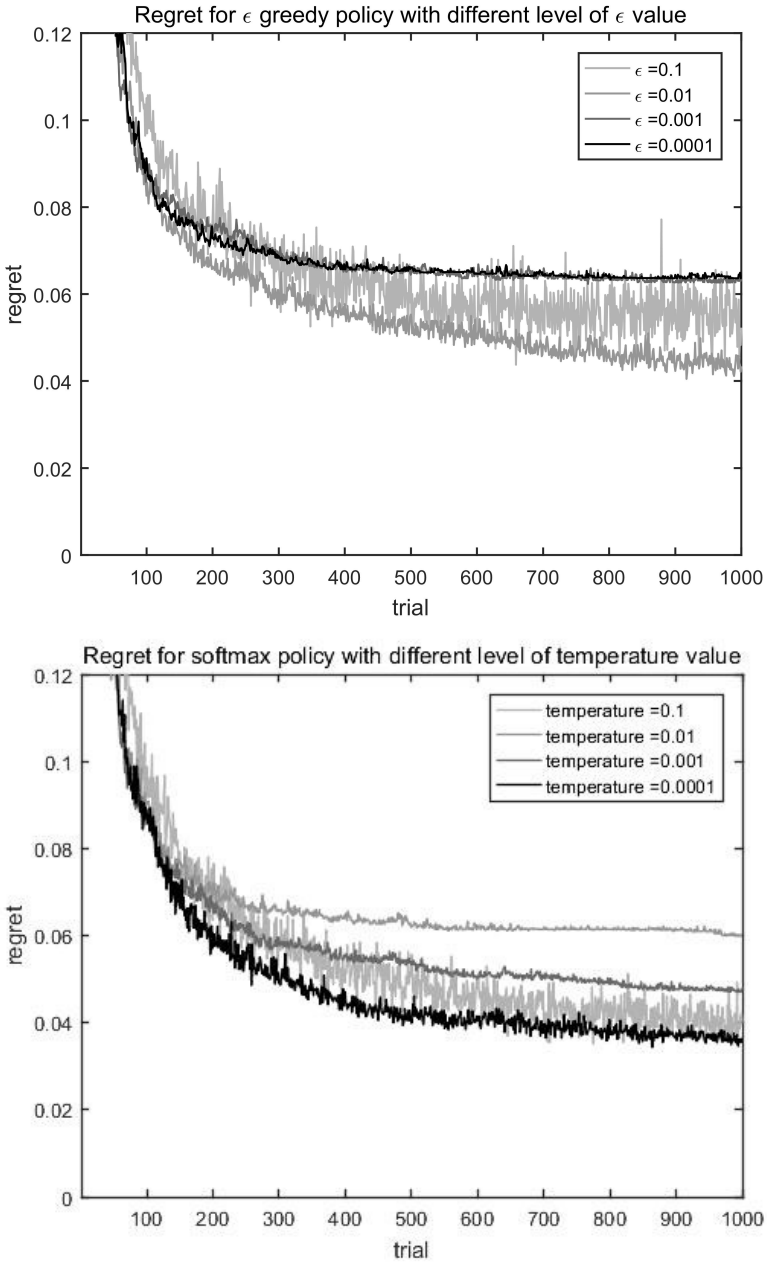

Fig. 3. Performance of each algorithm with different parameters ($k = 10$,$\sigma^2 = 1$): top–ε-greedy, bottom–Softmax

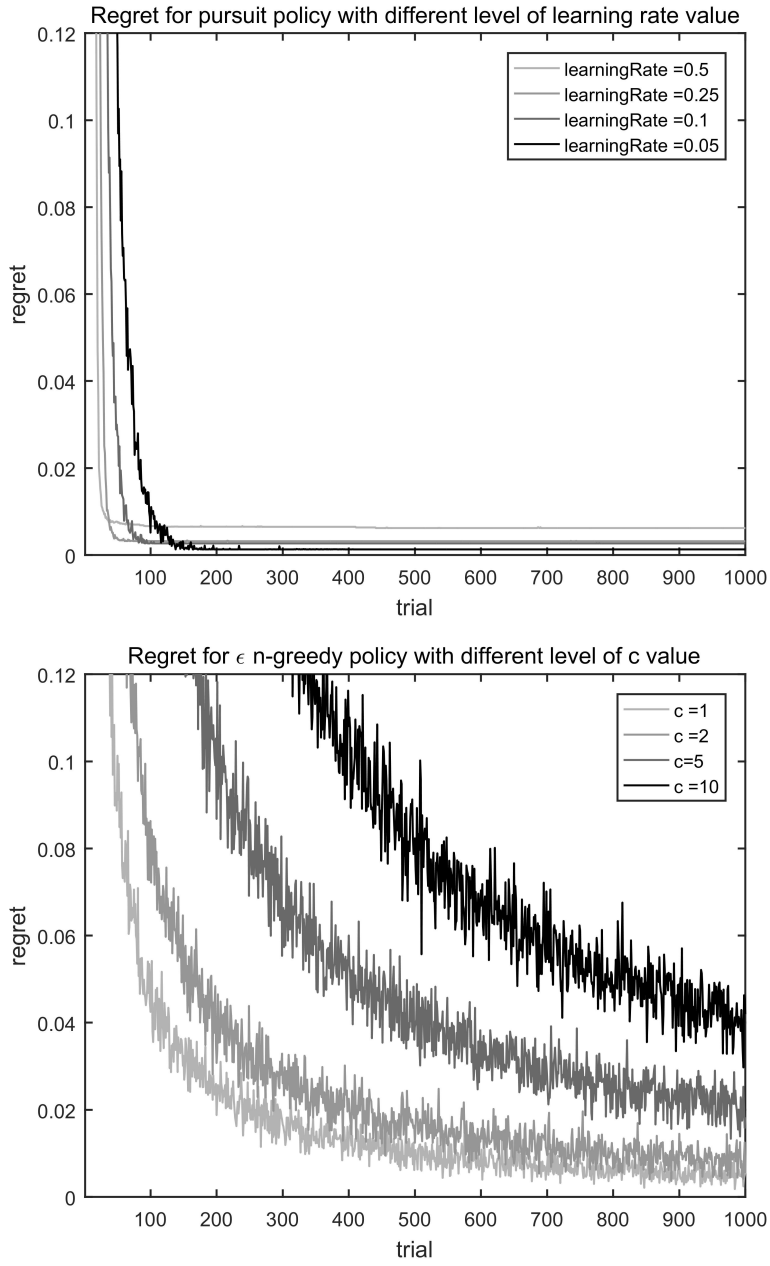

Fig. 4. Performance of each algorithm with different parameters ($k = 10$,$\sigma^2 = 1$): top–pursuit, bottom–ε_{n}-greedy

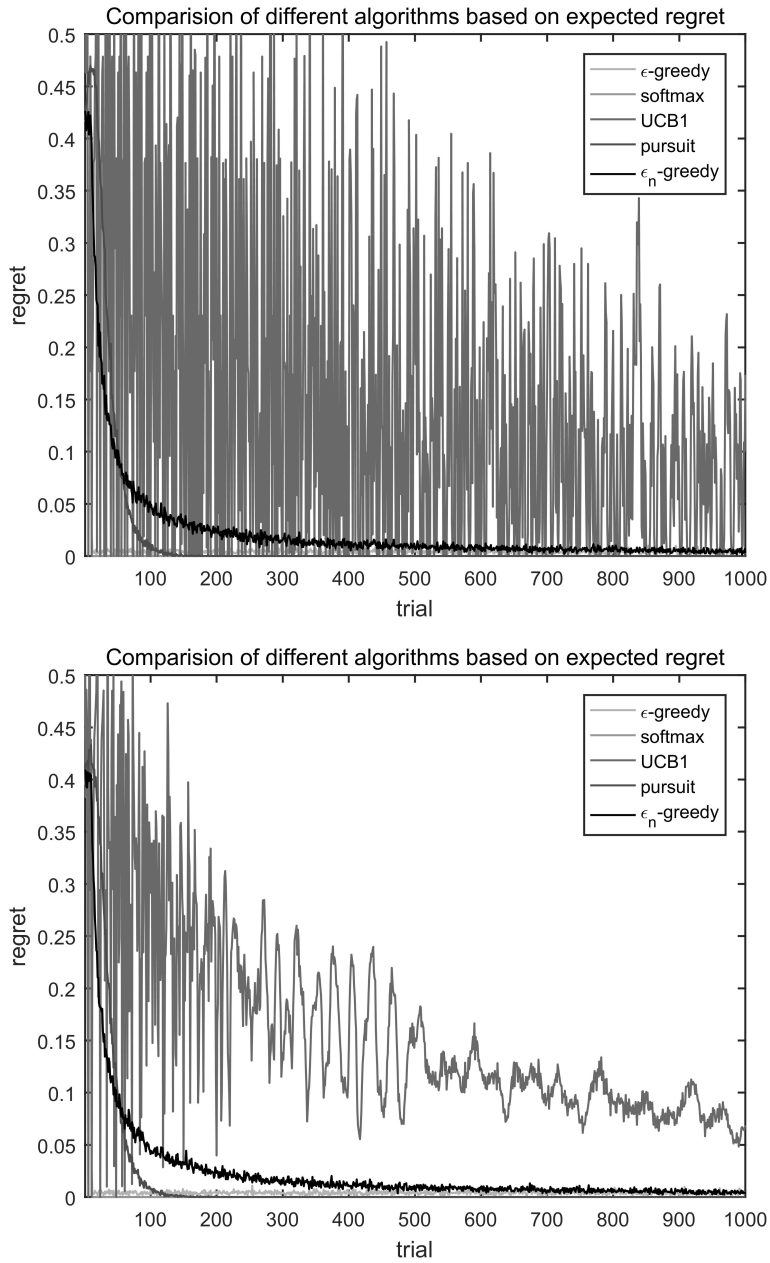

Fig. 5. Comparison of different algorithms with $K = 5$. For ε-greedy, set $\varepsilon = 0.01$, for Softmax, set learning rate as 0.01, for Pursuit, set temperature as 0.05, for ε_{n}-greedy, set $c = 1$: top–$k = 5$, $\sigma^2 = 0.001$; bottom–$k = 5$, $\sigma^2 = 0.01$

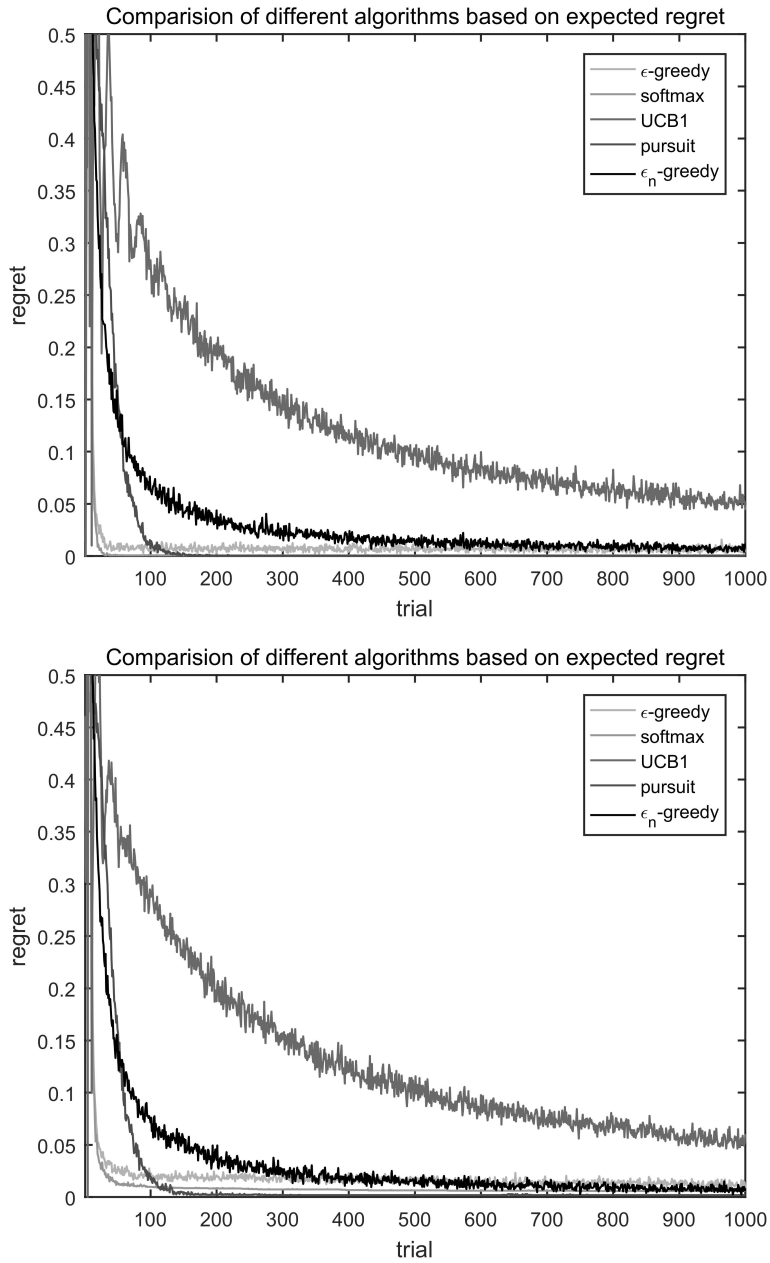

Fig. 6. Comparison of different algorithms with $K = 5$. For ε-greedy, set $\varepsilon = 0.01$, for Softmax, set learning rate as 0.01, for Pursuit, set temperature as 0.05, for ε_{n}-greedy, set $c = 1$: top–$k = 5$, $\sigma^2 = 0.1$; bottom–$k = 5$, $\sigma^2 = 0.2$

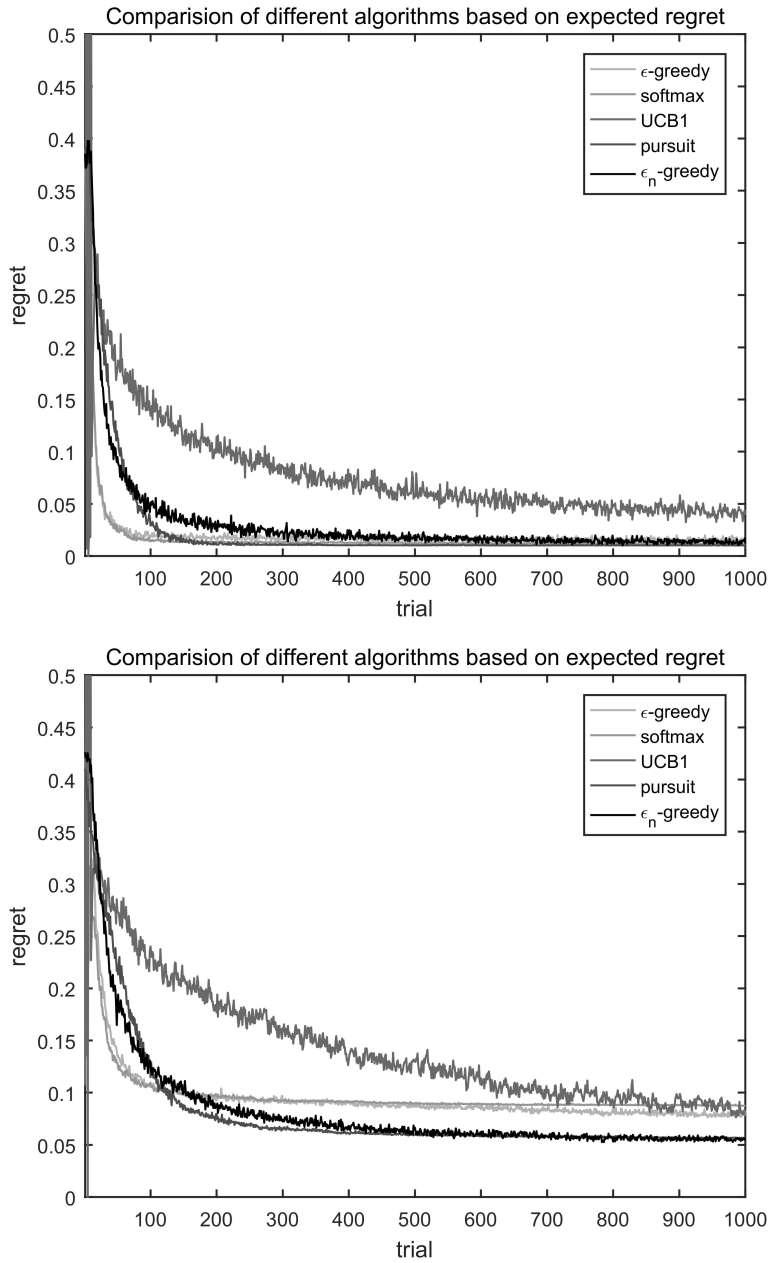

Fig. 7. Comparison of different algorithms with $K = 5$. For ε-greedy, set $\varepsilon = 0.01$, for Softmax, set learning rate as 0.01, for Pursuit, set temperature as 0.05, for ε_{n}-greedy, set $c = 1$: top–$k = 5$, $\sigma^2 = 0.5$; bottom–$k = 5$, $\sigma^2 = 0.8$

will have a lot of applications in real social life, such as stock selection strategies, balance between old and new listed companies' stock, or between old and new music recommendation.

As Fig. 8 shows, the dynamic performance of the ε-greedy and UCB1 algorithms are simulated. Run the algorithm 1000 trails. After 1000 trails, we add the same number of arms and continue with another 1000 trails. Finally, The accuracy of the algorithm is recorded in order to understand how they adapt to the dynamic situation. Set the number of k as 10, the variance as 0.2 and 0.5 respectively. In the simulation results, we can find that the ε-greedy algorithm is more stable and faster than the UCB1 algorithm in dynamic performance.

Finally, different algorithms will be used to select the optimal number of arms. Due to the different processing methods and the speed of different algorithms, the appropriate choice of the appropriate algorithm will contribute to the application efficiency.

As Figs. 9 and 10 show, the ε-greedy, ε_{n}-greedy , UCB1, Softmax and Pursuit algorithms are simulated for the same number of arms and different variance under their best parameters. The number of arms were set to 10, 20, 30 and 40, variance for the random generation. In this case, the ε-greedy algorithm is always able to converge and stabilize quickly, and Pursuit algorithm convergence rate is slower than the algorithm, but the many number of iterations can have a better degree of regret, the benefits may not increase, because there are too many best arms missed in the early stages.

5. Conclusion

In this paper, given a comprehensive evaluation of different multi armed algorithms, namely ε-greedy , ε_{n}-greedy, UCB1, Softmax and Pursuit. Each algorithm has been trailed in different situations and given a comparison of these five algorithms. From the trails, it can be found that the simplest algorithm performs the most stable and relatively low regret in both static and dynamic situations. Pursuit algorithm and Softmax algorithm second only to the ε-greedy algorithm, while the final regret on a slight advantage, but because of the slow convergence, will not necessarily be the best choice.

References

[1] R. Agrawal: *Sample mean based index policies with o (log n) regret for the multi-armed bandit problem.* Advances in Applied Probability *27* (1995), No. 4, 1054–1078.

[2] H. Robbinsh: *Some aspects of the sequential design of experiments.* Bulletin of the American Mathematical Society *58* (1952), No. 5, 527–535.

[3] J. C. Gittins: *Bandit processes and dynamic allocation indices.* Journal of the Royal Statistical Society. Series B (Methodological) *41* (1979), No. 2, 148–177.

[4] J. Vermorel, M. Mohri: *Multi-armed bandit algorithms and empirical evaluation.* European Conference on Machine Learning, 3–7 October 2005, Porto Portugal, Springer Nature, book series (LNCS) *3720* (2005) 437–448.

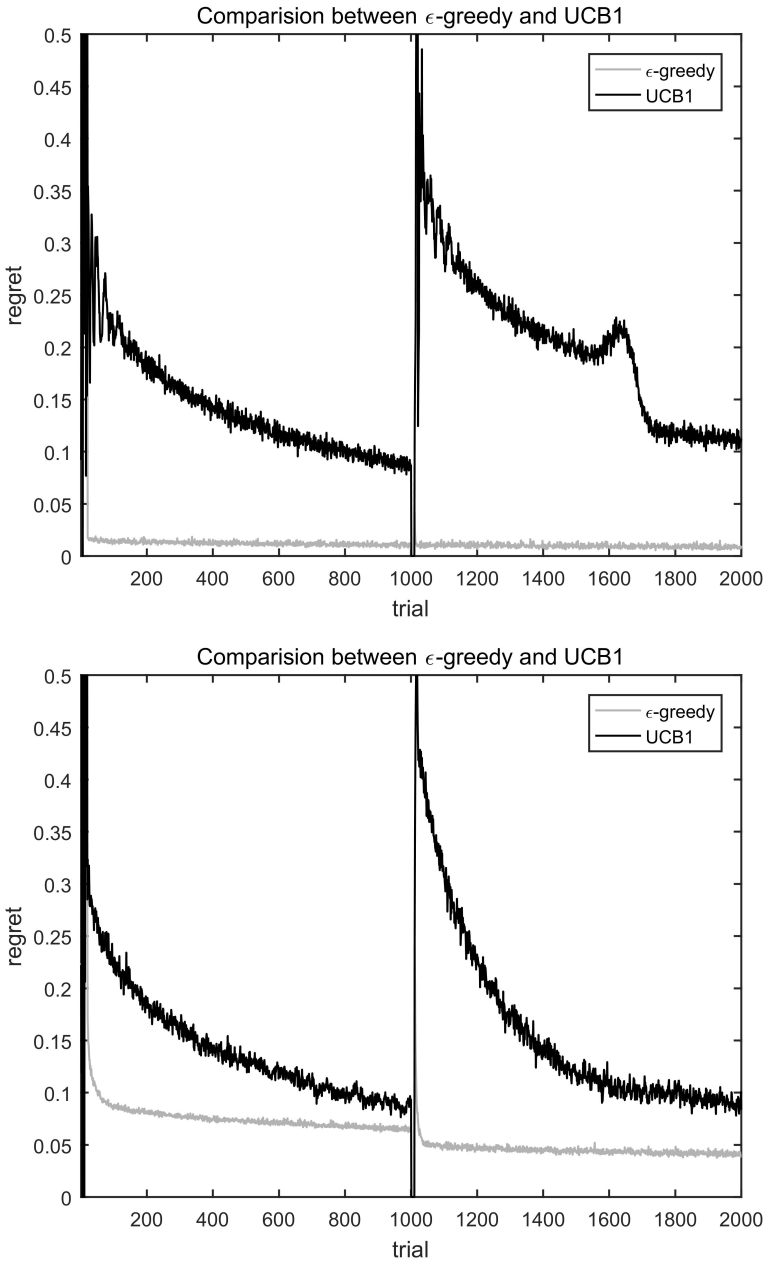

Fig. 8. Comparison between UCB1 and ε-greedy algorithms for dynamic multi-bandit problem: top–$k = 10$, $\sigma^2 = 0.2$; bottom–$k = 10$, $\sigma^2 = 0.5$

[5] P. Auer, N. Cesa-Bianchi, P. Fischer: *Finite-time analysis of the multiarmed ban-*

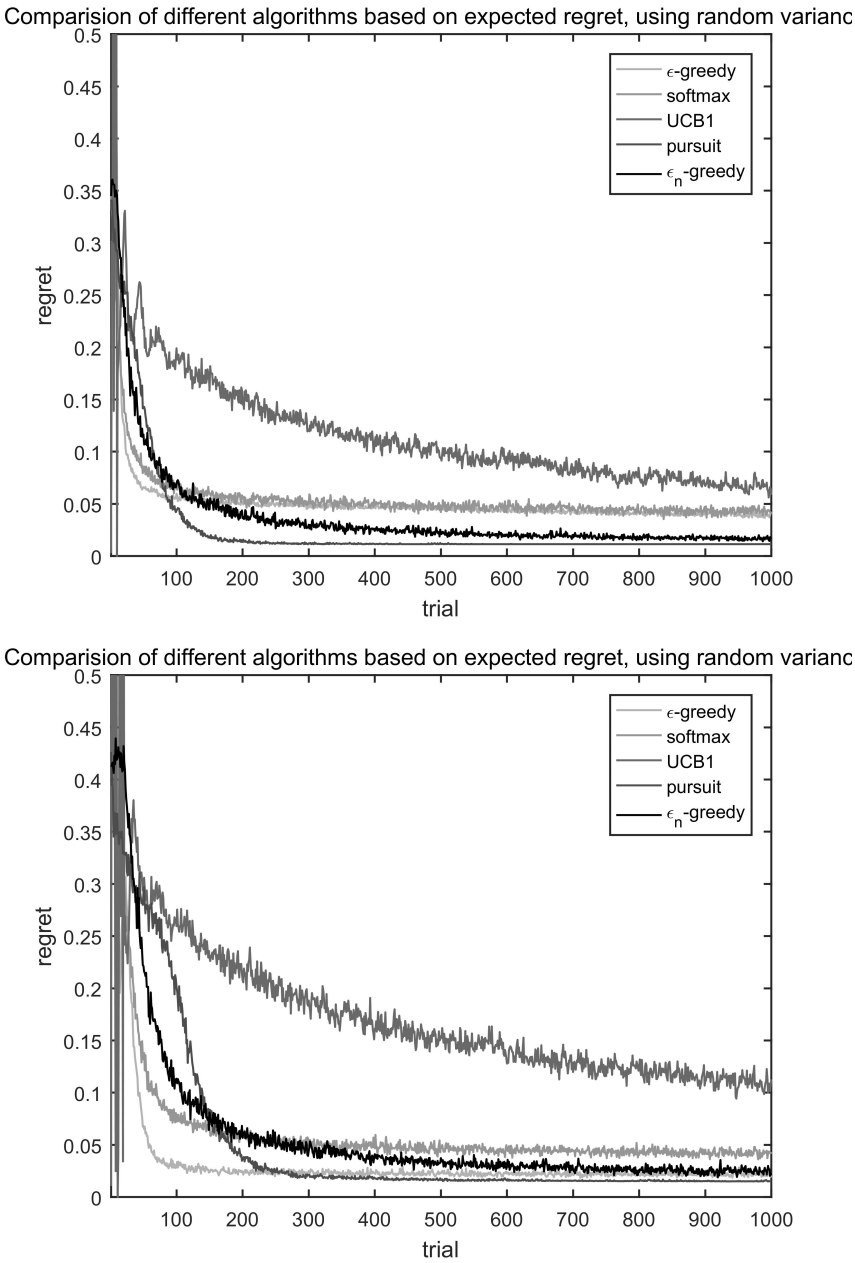

Fig. 9. Comparison between different algorithms for different K values, with random variances: top–$k = 10$; bottom–$k = 20$

dit problem. Machine Learning *47* (2002), Nos. 2–3, 235–256.

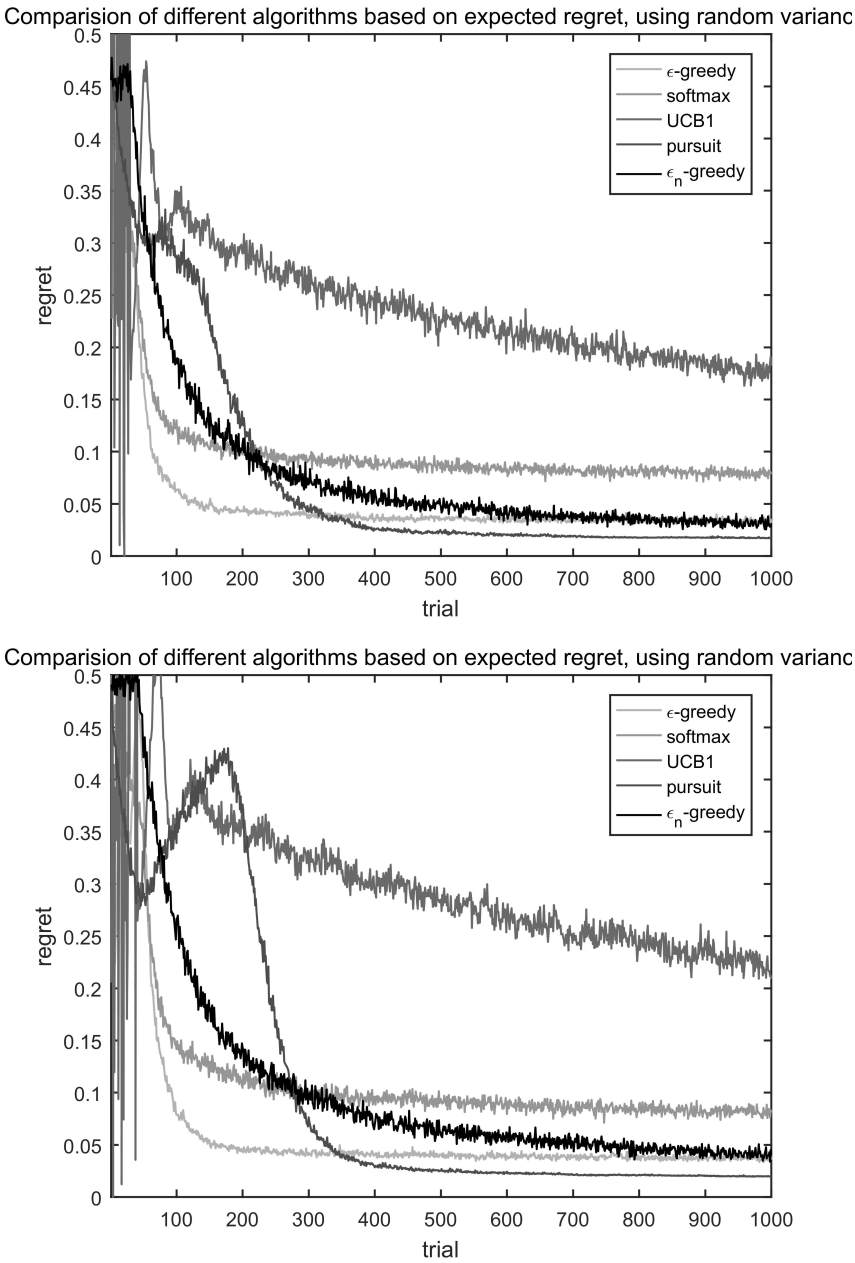

Fig. 10. Comparison between different algorithms for different K values, with random variances: top–$k = 30$; bottom–$k = 40$

[6] M. Tokic: *Adaptive ε-greedy exploration in reinforcement learning based on value dif-*

ferences. Annual Conference on Artificial Intelligence, KI 2010: Advances in Artificial Intelligence, 21–24 September 2010, Karlsruhe, Germany, Springer Nature, book series (LNCS) *6359* (2010), 203–210.

[7] P. Auer: *Using confidence bounds for exploitation-exploration trade-offs*. Journal of Machine Learning Research *3* (2003), 397–422.

[8] D. Bouneffouf, R. Féraud: *Multi-armed bandit problem with known trend*. Neurocomputing *205* (2016), 16–21.

[9] R. D. Luce: *Individual choice behavior: A theoretical analysis*. Journal of the American Statistical Association, Reprint of the John Wiley & Sons, Inc., New York (1959).

[10] M. A. L. Thathachar, P. S. Sastry: *A class of rapidly converging algorithms for learning automata*. IEEE International Conference on Cybernetics and Society, Bombay, India (1984), 602–606.

Received April 30, 2017