Overtaking Method based on Variance of Values: Resolving the Exploration–Exploitation Dilemma

Kento Ochi\textsuperscript{a}, Moto Kamiura\textsuperscript{a,b,*}

\textsuperscript{a}Tokyo Denki University, Graduate School of Science and Engineering, Ishizaka, Hatoyama-cho, Hiki-gun, Saitama, 350-0394, Japan
\textsuperscript{b}Research Institute of Electrical Communication, Tohoku University, 2-1-1 Katahira, Aoba-ku, Sendai 980-8577, Japan

Abstract

The exploration–exploitation dilemma is an attractive theme in reinforcement learning. Under the tradeoff framework, a reinforcement learning agent must cleverly switch between exploration and exploitation because an action, which is estimated as the best in the current learning state, may not actually be the true best. We demonstrate that an agent can determine the best action under certain conditions even if the agent selects the exploitation phase. Under the conditions, the agent does not need an explicit exploration phase, thereby resolving the exploration–exploitation dilemma. We also propose a value function on actions and how to update this value function. The proposed method, the “overtaking method,” can be integrated with existing methods, UCB1 and UCB1-tuned, for the multi-armed bandit problem without compromising features. The integrated models show better results than the original models.

© 2013 The Authors. Published by Elsevier B.V. Open access under CC BY-NC-ND license.
Selection and peer-review under responsibility of the Program Committee of IES2013

Keywords: Reinforcement learning; Exploration–exploitation dilemma; Variance; Overtaking method

1. Introduction

Reinforcement learning is a type of machine learning that is based on the maximization of total reward\textsuperscript{1}. A reinforcement learning agent adapts to an environment through behavioral trial and error, which are determined on...
the basis of the policy of the agent. In supervised learning, a system is given a desired output (i.e., supervisory signals). On the other hand, a reinforcement learning agent can know only a reward for each behavior, which is indirect information about the environment. Therefore, policy against the uncertainty of the environment is essential in reinforcement learning.

The uncertainty of the environment manifests as the exploration–exploitation dilemma, which is a decision tradeoff of the agent, i.e., search for a better action (exploration) or take a temporally selected action as the current optimal solution (exploitation). The exploration–exploitation dilemma has been well researched, and many models that address this dilemma have been proposed\textsuperscript{2,3,4}.

The dilemma is based on inconsistency between explorative action and exploitative action; taking a temporally selected action (exploitation) and searching for a better action (exploration) are incompatible. However, this dilemma is not absolute. When outputs of a value function for all actions are higher than the true values of averages, exploitative action can be consistent with explorative action.

Here we propose a new method based on the above perspective to resolve the dilemma, which we call the Overtaking method. The Overtaking method can be integrated into existing methods. In addition, we show that, when using the proposed method, integrated models demonstrate a better performance.

2. Reinforcement Learning

2.1. Conditions to accomplish best action in exploitation phase

Here we discuss the conditions required to accomplish the best action in the exploitation phase.

In the value function approach of reinforcement learning, given a state $s$ and an action $a$, the value $Q^\pi(s,a)$ for a policy $\pi$ is defined. There are many types of policies and value functions in existing methods. Greedy is one such policy, in which an agent selects an action for a state subject to the maximum value of $Q$.

In the simplest sense, given a state $s$ and an action $a$, a value function can be defined by a conditional expectation,

$$Q(s,a) := E[R \mid s,a],$$

where $R$ is a random variable that represents a reward from the environment. However, it is non-trivial whether an expectation is suitable for a value for an action. Generally, we can construct a value function that is not equal to an expectation.

To simplify, we assume the multi-armed bandit problem in which we can ignore the state of the agent. An action $a$ indicates the selection of the arm $a$. Let $\mu_a$ be an expected reward for an action $a$, and $Q_{a,n}$ be a value function, where $n$ is the number of times action $a$ is taken. Here the agent never knows the expected rewards. The value $Q_{a,n}$ is not the same as an expectation and is updated by taking action $a$. Using this situation, we can prove the following theorem.

**Theorem 1.** Assume a multi-armed bandit problem in which there are expected rewards $\mu_a$ for each action $a \in S_{\text{action}}$, where $S_{\text{action}}$ is a set of actions for the problem. If a value function $Q_{a,n}$ of a reinforcement learning agent with a greedy policy fulfills the following conditions (1) and (2), then the agent can select the best action in the exploitation phase, where $n \in \mathbb{N}$ is the number of times action $a$ is taken.

1. $Q_{a,n} \geq \mu_a$ for all $a \in S_{\text{action}}$ and all $n \in \mathbb{N}$

2. $\lim_{n \to \infty} Q_{a,n} = \mu_a$ for all $a \in S_{\text{action}}$
Proof

It is only necessary to prove the case of two actions. For \(a, b \in S_{\text{action}}\), assume two expected rewards \(\mu_a\) and \(\mu_b\), where \(\mu_b < \mu_a\). Therefore, the action that should be selected is \(a\). For \(\mu_b < \mu_a\) and condition (1), we consider three cases with permutations of the values \(Q_{a,a}\) and \(Q_{b,m}\), and the expected rewards, i.e., (i) \(\mu_b < \mu_a < Q_{b,m} < Q_{a,a}\), (ii) \(\mu_b < \mu_a < Q_{a,a} < Q_{b,m}\), and (iii) \(\mu_b < Q_{b,m} < \mu_a < Q_{a,a}\).

Given (i) \(\mu_b < \mu_a < Q_{b,m} < Q_{a,a}\), the agent selects action \(a\) until \(\mu_b < \mu_a < Q_{a,a} < Q_{b,m}\). Therefore, we obtain the following.

\[
\mu_b < \mu_a < Q_{b,m} < Q_{a,a} \quad \text{or} \quad \mu_b < \mu_a < Q_{a,a} \quad \text{or} \quad \mu_b < Q_{b,m} < \mu_a < Q_{a,a}
\]  

The right-side sequence is for case (ii).

Given (ii) \(\mu_b < \mu_a < Q_{a,a} < Q_{b,m}\), the agent selects action \(b\) until \(\mu_b < \mu_a < Q_{b,m} < Q_{a,a}\). Therefore, we obtain the following.

\[
\mu_b < \mu_a < Q_{a,a} < Q_{b,m} \quad \text{or} \quad \mu_b < \mu_a < Q_{b,m} < Q_{a,a}
\]  

The right-side sequence is for case (i). From (2), (3), and (4), we obtain the following.

\[
\mu_b < \mu_a < Q_{a,a} < Q_{b,m} \quad \text{or} \quad \mu_b < \mu_a < Q_{b,m} < Q_{a,a}
\]  

In this situation, the agent can select both actions \(a\) and \(b\). If the agent selects \(a\), then the selection is correct. If the agent selects \(b\), then the right-side sequence of (5) is as follows.

\[
\mu_b < Q_{b,m} < \mu_a = Q_{a,a}
\]  

Consequently, the agent can select action \(a\) for cases (i) and (ii).

Given (iii) \(\mu_b < Q_{b,m} < \mu_a < Q_{a,a}\), the agent selects action \(a\). From (2), case (iii) can proceed to (6) under the condition \(n \to \infty\). Thus, the agent can select action \(a\) for case (iii). [Q.E.D.]

2.2. Over-take expected reward model: OT model

Here we propose a method that enhances the probability of fulfilling conditions (1) and (2).

In Theorem 1, a value \(Q_{a,a}\) must exceed an expected reward \(\mu_a\) to select an action \(a\). One of the simplest countermeasures for this is to make each initial \(Q_{a,0}\) value very high, which enhances the probability that condition (1) will be fulfilled. However, significant time would be required to obtain (6) using this countermeasure. Therefore, we must estimate a reasonable value for \(Q_{a,a}\).

We focus on the central limit theorem. For cases in which the number of trials \(n\) is sufficient, the central limit theorem shows that the sample average follows normal distribution \(N(\mu_a, \frac{\sigma^2_a}{n})\). Let \(\mu_a\) be an expected reward for an action \(a\), and \(\sigma^2_a\) be the population variance of the reward. From these theorems, we propose a new model, the
Over-take expected reward model (OT model). The OT model determines the arbitrary probability to fulfill condition (1).

In the trial \( n \), the value function of action \( a \) is expressed as follows.

\[
Q_{a,n} = \text{Ave}_{a,n} + c \sqrt[2]{\frac{\text{Var}_{a,n}}{n_a}} \tag{7}
\]

\( \text{Ave}_{a,n} \) is the average for an action \( a \) at trial \( n \). Similarly, \( \text{Var}_{a,n} \) is the variance, and \( c \) is a real number parameter that determines the acceptable outlier range.

The convergence proof in this model is obtained easily.

\[
\lim_{n \to \infty} \left( \text{Ave}_{a,n} + c \sqrt[2]{\frac{\text{Var}_{a,n}}{n_a}} \right) = \mu_a + c \sqrt[2]{\frac{\sigma_a^2}{\infty}} = \mu_a \tag{8}
\]

(8) indicates that the proposed model fulfills condition (2).

\[
c \sqrt[2]{\frac{\text{Var}_{a,n}}{n_a}}
\]

is the standard deviation of reward average distribution (except that it uses current reward variance rather than population variance). This defines the acceptable range for the value taken by the value function and is determined by parameter \( c \).

The reason this model almost always fulfills condition (1) can be described as follows.

From the relational expression (1), \( E_a[\text{r}] \leq Q_{a,n} \).

Therefore, we must fulfill

\[
Q_{a,n} - E_a[\text{r}] \geq 0
\]

\[
\text{Ave}_{a,n} + c \sqrt[2]{\frac{\text{Var}_{a,n}}{n_a}} - E_a[\text{r}] \geq 0
\]

\[
\text{Ave}_{a,n} \geq E_a[\text{r}] - c \sqrt[2]{\frac{\text{Var}_{a,n}}{n_a}} \tag{9}
\]

The probability to fulfill (9) can be calculated using the central limit theorem.
This probability is uniquely determined by variance under the assumption that \( \text{Var}_{a,n} = \sigma_a^2 \), i.e., the proposed model can determine the probability of fulfilling condition (1) for each action. Therefore, the proposed model can ensure that the agent will eventually select the best action without selecting actions that do not need to be searched. This approach differs from existing models, which require that the selected probability is always greater than zero for each action. This condition also ensures that the agent will eventually select the best action. However, this condition allows the agent to search actions that do not need to be searched. On the other hand, this condition is not required in the proposed model.

It should be noted that we assume the variance, which is calculated by the reward obtained, is equal to population variance. However, we have not confirmed that this is indeed the case.

### 2.3. Parameter \( c \)

Here we examine the effect on the accuracy rate of different values for parameter \( c \).

![Fig. 1. Accuracy rate with varying values for parameter \( c \): (a) from the beginning; (b) after 150 plays](image)

Figure 1 shows the difference in the accuracy rate with varying values for parameter \( c \) using the proposed model. The simulation setting conforms to Simulation No.5 (describe at Chapter 4). The timing at which parameter \( c \) was set is different in Figs. 1(a) and 1(b). For Fig. 1(a), we set parameter \( c \) from the beginning, and for Fig. 1(b), we set parameter \( c \) after 150 plays (\( c \) was zero before 150 plays).

In both Figs. 1(a) and 1(b), accuracy rate increases as \( c \) increases. However, the increment of the accuracy rate decreases as \( c \) increases. In addition, it requires significant time for the value function to converge the expected reward when \( c \) is large. Thus, we must avoid setting parameter \( c \) to a value that is too large. In contrast, the accuracy rate is low when \( c \) is less than zero, because the probability of fulfilling (1) is low.

In addition, interesting results can be seen in Fig. 1(b). When \( c \) is changed in the middle of an episode, the accuracy rate temporarily decreases. This indicates that the agent selects another action whose expected reward is potentially higher. From these results, we recommend the early setting of parameter \( c \).

### 2.4. Computational complexity

In reinforcement learning, the simulation takes a significant amount of time and treats many states. Therefore, the calculation speed of the value function must be fast, and a moderate amount of memory is recommended. The proposed model uses both average and variance, and the calculation of average and variance requires each reward
sample if we use the original formula, i.e., the proposed model requires enormous amounts of memory and computation as the volume of data increases. To avoid this problem, we use a recurrence formula.

**Average**

\[
Ave_n = \frac{1}{n} \sum_{i=1}^{n} r_i \\
= Ave_{n-1} + \frac{1}{n} (r_n - Ave_{n-1}) 
\]

(11)

**Unbiased variance**

\[
Var_n = \frac{1}{n-1} \sum_{i=1}^{n} (r_i - Ave_n)^2 \\
= \frac{n-2}{n-1} Var_{n-1} + \frac{1}{n} (r_n - Ave_{n-1})^2 
\]

(12)

By using (11) and (12), we can solve both memory usage and computational complexity issues.

### 3. Integration with UCB1 and UCB1-tuned

The proposed model only adds one section using variance; thus, it is easy to integrate with existing models and does not require significant alteration of existing formulas. To demonstrate integration, we have integrated the proposed method into UCB1 and UCB1-tuned.

UCB1, which was proposed by Auer et al.\(^2\), attempts to select an action with a small number of trials. The formula is expressed as follows.

\[
Q_{a,n} = Ave_{a,n} + \sqrt{\frac{2 \ln n}{n_a}} 
\]

(13)

UCB1 selects an action by average reward and the number of trials and does not consider the variation of rewards.

UCB1-tuned, which was also proposed by Auer et al.\(^2\), extends UCB1 to consider variance. The formula is expressed as follows.

\[
Q_{a,n} = Ave_{a,n} + \sqrt{\frac{\ln n}{n_a} \min \left\{ \frac{1}{4}, \frac{1}{n_a} \sum_{i=1}^{n_a} r_{a,i}^2 - Ave_{a,n}^2 + \sqrt{\frac{2 \ln n}{n_a}} \right\}} 
\]

(14)

In UCB1-tuned, there is a situation that is similar to that observed when \( c \) is less than 1 in our model due to the product of variance and trial section.

We integrate the proposed method into these models.

\[
Q_{a,n} = Ave_{a,n} + c \sqrt{\frac{Var_{a,n}}{n_a}} + \sqrt{\frac{2 \ln n}{n_a}} 
\]

(15)
Similar to the original UCB1 model, the integrated models can select the action with a small number of trials, and similar to the OT model, the integrated models can always fulfill condition (1). We refer to (15) as “UCB1:OT” and (16) as “UCB1-tuned:OT.”

4. Experiments

Here we compare integrated models and the original models using the multi-armed bandit problem. In this problem, each bandit returns the reward following normal distribution to control the variation of rewards. The problem also follows normal distribution for population mean and population variance on the reward distribution of each bandit. This enables the comparison of the differences of variance for each action.

We tracked the rate at which the optimal bandit (accuracy rate) was selected. In each simulation, the number of bandit selections (plays) was 2000 in one episode, and the score was the average of 100000 episodes. An episode is an independent learning task, and the content of learning is not shared in each episode. To examine the performance of using variance, each action was selected two times with the first few plays.

In these experiments, parameter \( c \) was set to a fixed value, \( c = 1 \). Table 1 shows the settings for each simulation.

- Sim.1: basic simulation: population mean and variance generated with a standard normal distribution
- Sim.2: increased overall variance
- Sim.3: increased variation for each action
- Sim.4: increased variation on the average of each action
- Sim.5: integrate Sim.2, Sim.3, and Sim.4

<table>
<thead>
<tr>
<th>Simulation No.</th>
<th>Bandit</th>
<th>Average distribution</th>
<th>Variance distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>30</td>
<td>( N(0, 1) )</td>
<td>( N(0, 1) )</td>
</tr>
<tr>
<td>2</td>
<td>30</td>
<td>( N(0, 1) )</td>
<td>( N(10, 1) )</td>
</tr>
<tr>
<td>3</td>
<td>30</td>
<td>( N(0, 1) )</td>
<td>( N(0, 100) )</td>
</tr>
<tr>
<td>4</td>
<td>30</td>
<td>( N(0, 5) )</td>
<td>( N(0, 1) )</td>
</tr>
<tr>
<td>5</td>
<td>30</td>
<td>( N(0, 5) )</td>
<td>( N(10, 100) )</td>
</tr>
</tbody>
</table>

In each simulation, we compared SampleAverage and the propose model using \( \epsilon \)-Greedy and softmax policies. We also compared UCB1 and UCB1-tuned models and their respective integrated models.

The SampleAverage method uses the current average of rewards, and its value function is expressed as follows.

\[
Q_{a,n} = \text{Ave}_{a,n}
\]  

(17)

The \( \epsilon \)-Greedy policy selects greedy in most cases. However, with probability \( \epsilon \), it selects randomly. The softmax policy evaluates selection probability for each action, and its equation for selection probability is expressed as follows.

\[
P_{a,n} = \frac{\exp\left(\frac{Q_{a,n}}{\tau}\right)}{\sum_{i=1}^{\text{bandit}} \exp\left(\frac{Q_{i,n}}{\tau}\right)}
\]  

(18)
\( \tau \) is the positive parameter that balances exploration and exploitation. As \( \tau \) increases, this policy more closely approximates a greedy policy. The \( \epsilon \)-Greedy policy differs from the softmax policy in that the selection method of the softmax policy is probabilistic; thus, it is possible to select non-greedy actions using this policy.

In each simulation, we used reasonable values, i.e., \( \epsilon = 0.1 \) and \( \tau = \frac{1}{1 + 100 / \pi} \), to focus on the difference between the proposed and existing models.

4.1. Accuracy rate

Fig. 2. Accuracy rate for Simulation No.1

Fig. 3. Accuracy rate for Simulation No.2

Fig. 4. Accuracy rate for Simulation No.3
Table 2 shows the difference of accuracy rate between the integrated models and the existing models. The states of UCB1 and UCB1-tuned show the improvement of the accuracy rate by integration.

<table>
<thead>
<tr>
<th>Method</th>
<th>Sim1 [%]</th>
<th>Sim2 [%]</th>
<th>Sim3 [%]</th>
<th>Sim4 [%]</th>
<th>Sim5 [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>ε Greedy</td>
<td>+6.36</td>
<td>+11.4</td>
<td>+11.1</td>
<td>+3.47</td>
<td>+8.70</td>
</tr>
<tr>
<td>Softmax</td>
<td>-0.79</td>
<td>+7.74</td>
<td>+5.43</td>
<td>-0.19</td>
<td>+6.12</td>
</tr>
<tr>
<td>UCB1</td>
<td>-2.46</td>
<td>+5.94</td>
<td>+3.78</td>
<td>-0.76</td>
<td>+7.20</td>
</tr>
<tr>
<td>UCB1-tuned</td>
<td>+3.56</td>
<td>+15.7</td>
<td>+14.8</td>
<td>+2.25</td>
<td>+13.0</td>
</tr>
</tbody>
</table>

The experimental results can be summarized as follows. The proposed model achieved better results than the existing model when the variance of the reward was large. However, the proposed model achieved poorer results than softmax and UCB1 when the variance was small. We speculate that this was due to the stochastic reward policy implemented in softmax. We designed the OT method such that it was suitable for use with a greedy policy. Therefore, the proposed model is not suitable for use with a stochastic policy. However, it is not clear why the proposed model did not perform as well when integrated into UCB1. This result may be because the number of trials is significant for the value function, which is also true for UCB1-tuned. Nevertheless, performance was improved. However, further clarification is required.

We developed a hypothesis to explain why the proposed model and the integrated model achieved better results when the variance was large. We postulate that these results occurred, because the difference between the average outlier and the expected value increased proportionally with the reward variance. The agent may not select an outlier action as the variance increases, and consequently, the accuracy rate decreases in the existing models. The OT model ensures that the agent selects the best action regardless of the variance. As a result, the proposed model would achieve a higher accuracy rate than the existing models.
5. Summary and conclusions

We focused on a condition to determine the best action even with a greedy selection policy. We tested condition (1) and proposed a new model that fulfills the condition. We integrated the proposed model with existing models: UCB1 and UCB1-tuned. In addition, we assessed the integrated models and showed that they gave better results than the original models.

As mentioned in the introduction, resolving the exploration–exploitation dilemma is a crucial research problem. Existing methods ensure that the agent finds the best action. However, in existing methods, the selection probability must always be greater than zero for all actions. Existing methods deal with the problem by reducing the selection probability. We verified the effectiveness of a condition that enables the agent to select the best action and set the selection probability of non-best actions to zero. The proposed condition allows the agent to budget learning time for the selectable actions. Existing methods do not consider the possibility that the agent selects all actions. In contrast, the condition we proposed allows the agent to eliminate non-best actions. The proposed condition, which searches only for necessary actions, is a crucial approach for resolving the exploration–exploitation dilemma.

We determined that absolutely fulfilling the condition by the uncertainty of the environment is difficult. Therefore, we formulated a new model that almost always fulfilled the condition. The OT model calculates the probability distribution of the reward average by the central limit theorem and adds standard deviation and reward average, which accept a certain outlier level. In the proposed model, the probability to fulfill the condition is proportional to parameter $c$. As a result, the OT model ensures that the agent selects the best action even if the policy is fixed as exploitation.

From the experimental results, we identified a problem with the proposed model. It was that the proposed model was incompatible with a stochastic policy and a policy that used other factors with great weight for the value function. Selecting the best action requires a significant amount of time in these cases. As a result, the proposed model showed poorer performance than the existing models. We will address this problem in future research.

It is important to note that the following issues have not been considered in this study. We assumed that rewards followed normal distribution. Therefore, the average reward also followed normal distribution, although the number of trials was less. However, in the central limit theorem, when the value of sample $n$ is large, the sample average follows normal distribution. In other words, we assumed that the average reward follows normal distribution no matter how small $n$ is. In short, our assumption is not viable when the reward follows other distributions. In addition, we assumed that the reward variance was equal to the population variance. However, sample variance is usually different from population variance; therefore, future research will need to examine the extent to which our assumption is significant. Finally, we let $c$ be a constant. In future, we will need to examine comparative results for variable $c$ values.

Acknowledgements

Part of this work was carried out under the Cooperative Research Project Program of the Research Institute of Electrical Communication, Tohoku University.

References