Applied Soft Computing 70 (2018) 80–85

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Determining the optimal temperature parameter for Softmax function in reinforcement learning

Yu-Lin He[a,b,*], Xiao-Liang Zhang[a,b], Wei Ao[a,b], Joshua Zhexue Huang[a,b,*]

[a] Big Data Institute, College of Computer Science & Software Engineering, Shenzhen University, Shenzhen 518060, Guangdong, China
[b] National Engineering Laboratory for Big Data System Computing Technology, Shenzhen University, Shenzhen 518060, Guangdong, China

ARTICLE INFO

Article history:
Received 6 March 2018
Received in revised form 24 April 2018
Accepted 7 May 2018

Keywords:
Softmax function
Temperature parameter
Probability vector
Reinforcement learning
D-armed bandit problem

ABSTRACT

The temperature parameter plays an important role in the action selection based on Softmax function which is used to transform an original vector into a probability vector. An efficient method named Opti-Softmax to determine the optimal temperature parameter for Softmax function in reinforcement learning is developed in this paper. Firstly, a new evaluation function is designed to measure the effectiveness of temperature parameter by considering the information-loss of transformation and the diversity among probability vector elements. Secondly, an iterative updating rule is derived to determine the optimal temperature parameter by calculating the minimum of evaluation function. Finally, the experimental results on the synthetic data and $\mathcal{D}$-armed bandit problems demonstrate the feasibility and effectiveness of Opti-Softmax method.

1. Introduction

Softmax function is a normalized exponential function [4] which transforms a $\mathcal{D}$-dimensional original vector with arbitrary real values into a $\mathcal{D}$-dimensional probability vector with real values in the range [0, 1] that add up to 1. Softmax function is commonly applied to the fields of machine learning, such as logistic regression [5], artificial neural networks [15], reinforcement learning [17]. In general, Softmax functions without temperature parameters are used in the multi-class classification problem of logistic regression and the final layer of an artificial neural network, while Softmax function with temperature parameter [17] is used to convert the action rewards into the action probabilities in reinforcement learning.

The temperature parameter is an important learning parameter for the exploration-exploitation tradeoff in Softmax action selection. The large temperature parameter will lead to the exploration-only state (the actions have the almost same probabilities to be selected), while the small temperature parameter will result in the exploitation-only state (the actions with the higher rewards are more easily selected). This paper focuses on Softmax function-based exploration-exploitation tradeoff in the scenario of $\mathcal{D}$-armed bandit [21] which is a classical action selection problem of reinforcement learning. Some representative studies related to Softmax action selection are summarized as follows. Koulouriotis and Xanthopoulos in [12] examined Softmax algorithm with temperature parameter 0.3. Tokic and Palm in [19] tested the performances of Softmax action selection algorithms with temperature parameters 0.04, 0.1, 1, 10, 25 and 100. In [13], Kuleshov and Precup presented a thorough empirical comparison among the most popular multi-armed bandit algorithms, including Softmax function with temperature parameters 0.001, 0.007, 0.01, 0.05 and 0.1. Other studies with regard to Softmax action selection can be found in literatures [1,6,8,11,16,18]. To our best knowledge, the existing studies mainly used the trial-and-error strategy to select the temperature parameter for Softmax function when dealing with $\mathcal{D}$-armed bandit problem.

The simply Softmax function will be a very efficient action selection strategy to solve $\mathcal{D}$-armed bandit problem if the appropriate temperature parameter can be determined automatically. Up to now, there is no study that provides such automatic temperature parameter selection for Softmax function when dealing with $\mathcal{D}$-armed bandit problems. Thus, we develop a useful method named Opti-Softmax to determine the optimal temperature parameter for Softmax function in this paper. Firstly, we design a new evaluation function to measure the effectiveness of temperature parameter. The evaluation function includes two parts: the information-loss between the original vector and the probability vector and the diversity among probability vector elements. Secondly, we derive

* Corresponding authors at: Big Data Institute, College of Computer Science & Software Engineering, Shenzhen University, Shenzhen 518060, Guangdong, China.
E-mail addresses: yulinhe@szu.edu.com (Y.-L. He), zhangxlas@163.com (X.-L. Zhang), aowei2016@email.szu.edu.cn (W. Ao), zx.huang@szu.edu.cn (J.Z. Huang).

https://doi.org/10.1016/j.asoc.2018.05.012
1568-4946/

an iterative updating rule to determine the optimal temperature parameter by calculating the minimum of evaluation function. Finally, the necessary experiments on the synthetic data and $\mathcal{D}$-armed bandit problems are carried out to validate the performance of our proposed Opti-Softmax method.

The rest of the paper is organized as follows. Section 2 states the problem formulations of Softmax function. Section 3 gives the Opti-Softmax method to determine the optimal temperature parameter for Softmax function. Some experimental simulations are given in Section 4. Finally, Section 5 presents a brief conclusion to this paper.

2. Problem formulations of Softmax function

Given a $\mathcal{D}$-dimensional original vector $\vec{x} = (x_1, x_2, \ldots, x_{\mathcal{D}})$, $x_d \in \mathbb{R}$, $d = 1, 2, \ldots, \mathcal{D}$ and there exists $k \in \{1, 2, \ldots, \mathcal{D}\}$ such that $x_k \neq 0$, it can be transformed into a $\mathcal{D}$-dimensional probability vector $\vec{p} = (p_1, p_2, \ldots, p_{\mathcal{D}})$ with the following Softmax function:

$$p_d = \frac{\exp\left(\frac{x_d}{\tau}\right)}{\sum_{k=1}^{\mathcal{D}} \exp\left(\frac{x_k}{\tau}\right)}, \tag{1}$$

where $p_d \in (0, 1)$, $\sum_{d=1}^{\mathcal{D}} p_d = 1$ and $\tau > 0$ is the temperature parameter which has an important influence on the transformation performance of Softmax function. When $\tau \to +\infty$, $p_d \to \frac{1}{\mathcal{D}}$, i.e., the diversity among p_ds is small; when $\tau \to 0$, $|p_i - p_j| \gg |x_i - x_j|$, $\exists\, i, j \in \{1, 2, \ldots, \mathcal{D}\}$, $i \neq j$. i.e., the diversity among p_ds is large. Fig. 1 provides an example to show the influence of τ on Softmax function. 20 real numbers (blue bars) belonging to interval [0.1, 0.2] are randomly generated. In this figure, we can see that the probability vector elements corresponding to $\tau = 1$ (red bars) are almost $\frac{1}{20} = 0.05$, while $\tau = 0.01$ make some probability vector elements (green bars) be close to 0.

In reinforcement learning, Softmax function can be used to select the bandit-arm in $\mathcal{D}$-armed bandit, where each bandit-arm provides a random reward for gambler. Assume $\vec{x} = (x_1, x_2, \ldots, x_{\mathcal{D}})$ is the reward vector corresponding to $\mathcal{D}$ bandit-arms. Then, p_d is the probability with which the d-th bandit-arm is selected. $\tau \to +\infty$ will lead to the exploration-only state (the bandit-arms have almost the same probabilities to be selected), while $\tau \to 0$ will result in the exploitation-only state (the bandit-arms with the higher rewards are more easily selected). The key of solving $\mathcal{D}$-armed bandit problem is how to select the bandit-arms so that the gambler can obtain the maximal reward.

3. Determination of the optimal temperature parameter for Softmax function

This section presents a new method named Opti-Softmax to determine the optimal temperature parameter τ_{Opti} for Softmax function. The following evaluation function $L(\tau)$ is firstly designed to measure the effectiveness of temperature parameter:

$$L(\tau) = (\mathrm{H}_{\vec{z}} - \mathrm{H}_{\vec{p}})^2 + \lambda \mathrm{H}_{\vec{p}}^2, \tag{2}$$

where

$$\mathrm{H}_{\vec{z}} = -\sum_{d=1}^{\mathcal{D}} z_d \ln(z_d) \tag{3}$$

is the amount of information about $\vec{z} = (z_1, z_2, \ldots, z_{\mathcal{D}})$ which is the equivalent vector of $\vec{x}$, $z_d \in (0, 1)$, $\sum_{d=1}^{\mathcal{D}} z_d = 1$;

$$\mathrm{H}_{\vec{p}} = -\sum_{d=1}^{\mathcal{D}} p_d \ln(p_d) \tag{4}$$

is the amount of information about the probability vector $\vec{p} = (p_1, p_2, \ldots, p_{\mathcal{D}})$,

$$p_d = \frac{\exp\left(\frac{z_d}{\tau}\right)}{\sum_{k=1}^{\mathcal{D}} \exp\left(\frac{z_k}{\tau}\right)}, \tag{5}$$

$p_d \in (0, 1)$, $\sum_{d=1}^{\mathcal{D}} p_d = 1$; and $\lambda > 0$ is the enhancement factor.

The first term in Eq. (2) is to measure the information-loss after transforming the original vector $\vec{x}$ into the probability vector $\vec{p}$. Because $x_d \in \mathbb{R}$, $d = 1, 2, \ldots, \mathcal{D}$, we cannot obtain the information-amount of $\vec{x}$ directly. Hence, a linear transformation is performed on the original vector $\vec{x}$ and then generates its equivalent vector $\vec{z}$ as

$$z_d = \frac{x_d + 2|x_{\min}| + 0.01}{\sum_{k=1}^{\mathcal{D}} [x_k + 2|x_{\min}| + 0.01]}, \tag{6}$$

where $x_{\min} = \min\{x_1, x_2, \ldots, x_{\mathcal{D}}\}$. For the original vector $\vec{x} = (x_1, x_2, \ldots, x_{\mathcal{D}})$, $\exists\, k \in \{1, 2, \ldots, \mathcal{D}\}$, $x_k \neq 0$, Eq. (6) ensures $0 < z_d < 1$ for $\forall d \in \{1, 2, \ldots, \mathcal{D}\}$. The role of Eq. (6) is to facilitate the calculation of information-amount. We cannot calculate the information-amount of original vector directly if the elements of original vector are beyond the interval (0, 1). Thus, we need to transform the original vector into an equivalent vector in which the elements are all within the interval (0, 1). The second term in Eq. (2) is to control the diversity among probability vector elements $p_1, p_2, \ldots, p_{\mathcal{D}}$. $\mathrm{H}_{\vec{p}}$ attains its maximum $\ln(\mathcal{D})$ at $p_1 = p_2 = \cdots = p_{\mathcal{D}} = \frac{1}{\mathcal{D}}$. We hope that the optimal temperature parameter τ_{Opti} not only minimizes the information-loss of transformation but also maximizes the diversity among probability vector elements. Thus, we can get the optimality expression of τ_{Opti} as

$$\tau_{\text{Opti}} = \arg\min_{\tau > 0} L(\tau) = \arg\min_{\tau > 0} \left[(\mathrm{H}_{\vec{z}} - \mathrm{H}_{\vec{p}})^2 + \lambda \mathrm{H}_{\vec{p}}^2\right]. \tag{7}$$

Let $E = \sum_{d=1}^{\mathcal{D}} \exp\left(\frac{z_d}{\tau}\right)$ and $F = \sum_{d=1}^{\mathcal{D}} \left[\frac{z_d}{\tau} \exp\left(\frac{z_d}{\tau}\right)\right]$, $\mathrm{H}_{\vec{p}}$ in Eq. (4) can be equivalently written as

$$\begin{aligned}
\mathrm{H}_{\vec{p}} &= -\sum_{d=1}^{\mathcal{D}} \left[\frac{\exp\left(\frac{z_d}{\tau}\right)}{\sum_{k=1}^{\mathcal{D}} \exp\left(\frac{z_k}{\tau}\right)} \ln\left[\frac{\exp\left(\frac{z_d}{\tau}\right)}{\sum_{k=1}^{\mathcal{D}} \exp\left(\frac{z_k}{\tau}\right)}\right]\right] \\
&= -\sum_{d=1}^{\mathcal{D}} \left[\frac{\exp\left(\frac{z_d}{\tau}\right)}{\sum_{k=1}^{\mathcal{D}} \exp\left(\frac{z_k}{\tau}\right)} \left[\ln\left[\exp\left(\frac{z_d}{\tau}\right)\right] - \ln\left[\sum_{k=1}^{\mathcal{D}} \exp\left(\frac{z_k}{\tau}\right)\right]\right]\right] \\
&= -\frac{1}{E} \sum_{k=1}^{\mathcal{D}} \left[\exp\left(\frac{z_d}{\tau}\right) \left[\frac{z_d}{\tau} - \ln(E)\right]\right] \\
&= -\frac{F}{E} + \ln(E)
\end{aligned} \tag{8}$$

Bringing Eq. (8) into Eq. (2), we can obtain

$$L(\tau) = \left[\mathrm{H}_{\vec{z}} + \frac{F}{E} - \ln(E)\right]^2 + \lambda \left[\frac{F}{E} - \ln(E)\right]^2. \tag{9}$$

It is very difficult to determine the analytic formulation of τ_{Opti} by solving $\frac{dL(\tau)}{d\tau} = 0$. Because E and F are the functions with respect to τ, we try to find the optimal E or F which can minimize $L(T)$ in Eq. (9) by calculating

$$\begin{aligned}
\frac{dL(\tau)}{dE} &= 2\left[\mathrm{H}_{\vec{z}} + \frac{F}{E} - \ln(E)\right]\left(-\frac{F}{E^2} - \frac{1}{E}\right) + 2\lambda\left[\frac{F}{E} - \ln(E)\right]\left(-\frac{F}{E^2} - \frac{1}{E}\right) \\
&= -2\left(\frac{F+E}{E^2}\right)\left[\mathrm{H}_{\vec{z}} + (1+\lambda)\frac{F}{E} - (1+\lambda)\ln(E)\right] \\
&= 0
\end{aligned} \tag{10}$$

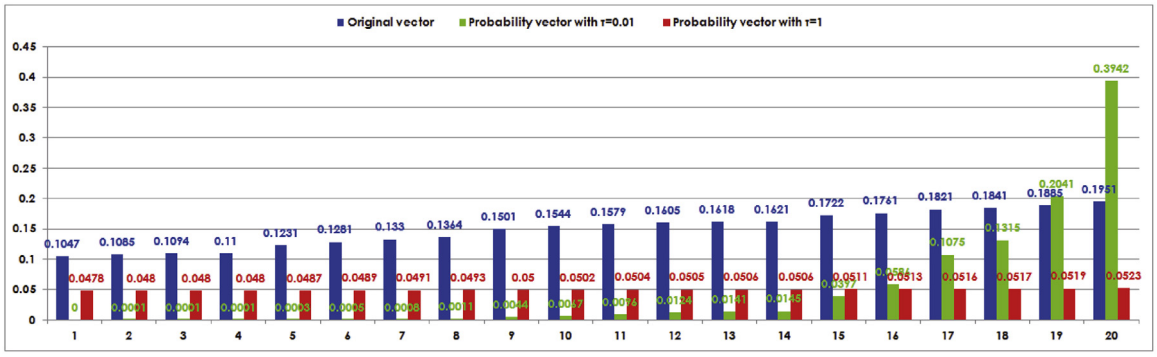


Fig. 1. The influence of τ on Softmax function. (For interpretation of the references to color in the text, the reader is referred to the web version of this article.)

or

$$\frac{dL(\tau)}{dF} = \frac{2}{E}\left[H_{\vec{z}} + \frac{F}{E} - \ln(E)\right] + \frac{2\lambda}{E}\left[\frac{F}{E} - \ln(E)\right] = \frac{2}{E}\left[H_{\vec{z}} + (1+\lambda)\frac{F}{E} - (1+\lambda)\ln(E)\right] = 0 \quad (11)$$

Because $E>0$ and $F>0$, we have

$$H_{\vec{z}} + (1+\lambda)\frac{F}{E} - (1+\lambda)\ln(E) = 0, \quad (12)$$

i.e.,

$$\sum_{d=1}^{\mathcal{D}}\left[\frac{z_d}{\tau}\exp\left(\frac{z_d}{\tau}\right)\right] = \left[\sum_{d=1}^{\mathcal{D}}\exp\left(\frac{z_d}{\tau}\right)\right]\left[\ln\left[\sum_{d=1}^{\mathcal{D}}\exp\left(\frac{z_d}{\tau}\right)\right] - \frac{H_{\vec{z}}}{1+\lambda}\right]. \quad (13)$$

Eq. (13) can be further simplified as

$$\tau = \frac{\sum_{d=1}^{\mathcal{D}}\left[z_d\exp\left(\frac{z_d}{\tau}\right)\right]}{\left[\sum_{d=1}^{\mathcal{D}}\exp\left(\frac{z_d}{\tau}\right)\right]\left[\ln\left[\sum_{d=1}^{\mathcal{D}}\exp\left(\frac{z_d}{\tau}\right)\right] - \frac{H_{\vec{z}}}{1+\lambda}\right]}. \quad (14)$$

Eq. (14) is the heuristic updating rule of Opti-Softmax method to determine the optimal temperature parameter τ_{Opti}. According to this updating rule, Opti-Softmax method as shown in Algorithm 1 gives an iterative procedure to determine τ_{Opti}.

Algorithm 1. Opti-Softmax method

Input: The original vector $\vec{x} = (x_1, x_2, \ldots, x_{\mathcal{D}})$, $x_d \in \mathbb{R}$, $d = 1, 2, \ldots, \mathcal{D}$ and $\exists\, k \in \{1, 2, \ldots, \mathcal{D}\}$, $x_k \neq 0$; the enhancement factor $\lambda > 0$; the stopping threshold $\xi > 0$; the initial temperature parameter $\tau_0 > 0$.
Output: The probability vector $\vec{p} = (p_1, p_2, \ldots, p_{\mathcal{D}})$, $p_d \in (0, 1)$, $d = 1, 2, \ldots, \mathcal{D}$, $\sum_{d=1}^{\mathcal{D}} p_d = 1$ and the optimal temperature parameter τ_{Opti}.
Calculating the equivalent vector $\vec{z} = (z_1, z_2, \ldots, z_{\mathcal{D}})$, $z_d \in (0, 1)$, $d = 1, 2, \ldots, \mathcal{D}$, $\sum_{d=1}^{\mathcal{D}} z_d = 1$ according to Eq. (6);
Calculating the information-amount $H_{\vec{z}}$ of $\vec{z}$ according to Eq. (3);
repeat
  $\tau_{\text{Opti}} = \tau_0$;
  $\tau_0 = \dfrac{\sum_{d=1}^{\mathcal{D}}\left[z_d\exp\left(\frac{z_d}{\tau_{\text{Opti}}}\right)\right]}{\left[\sum_{d=1}^{\mathcal{D}}\exp\left(\frac{z_d}{\tau_{\text{Opti}}}\right)\right]\left[\ln\left[\sum_{d=1}^{\mathcal{D}}\exp\left(\frac{z_d}{\tau_{\text{Opti}}}\right)\right] - \frac{H_{\vec{z}}}{1+\lambda}\right]}$;
until $\left|\tau_{\text{Opti}} - \tau_0\right| < \xi$
$\tau_{\text{Opti}} = \tau_0$;
$p_d = \dfrac{\exp\left(\frac{z_d}{\tau_{\text{Opti}}}\right)}{\sum_{k=1}^{\mathcal{D}}\exp\left(\frac{z_k}{\tau_{\text{Opti}}}\right)}$, $d = 1, 2, \ldots, \mathcal{D}$;

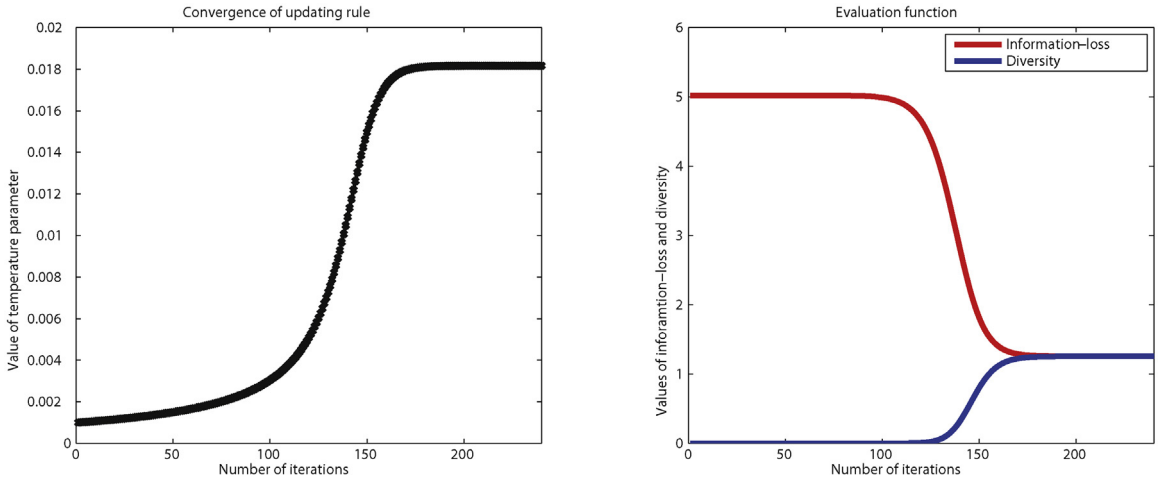
4. Experimental simulations

Two experiments are conducted to demonstrate the feasibility and effectiveness of Opti-Softmax method. The first experiment is to show that the updating rule as shown in Eq. (14) is convergent and the second experiment is to use Opti-Softmax method to deal with the $\mathcal{D}$-armed bandit problems in reinforcement learning.

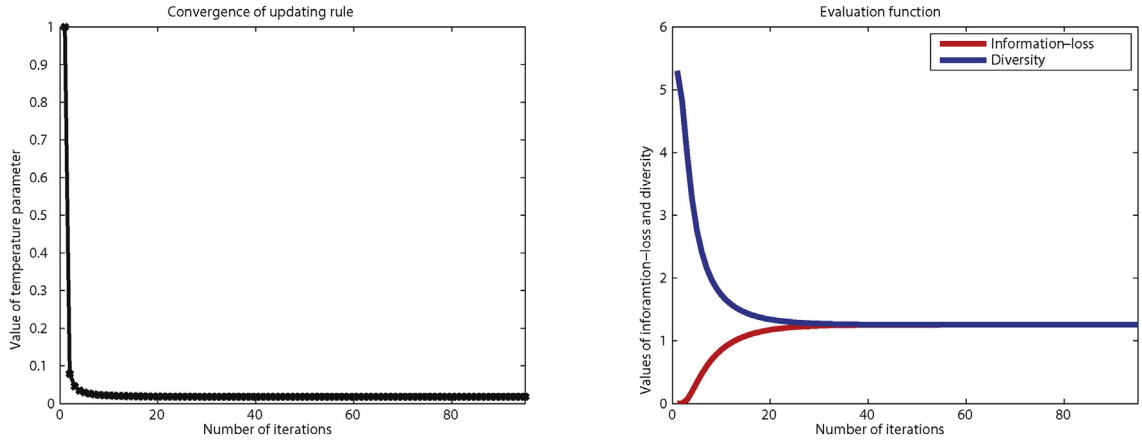
In the first experiment, a 10-dimensional original vector[1] $\vec{x} = (-7013.7933, -7282.7121, 649.9646, 4515.7862, -2025.9390, -2831.6297, -4294.4118, 7372.7049, 2528.2535, -5176.5538)$ is randomly generated in the interval $[-10000, 10000]$. We test the working performances of Opti-Softmax method (the enhancement factor $\lambda = 1$ and the stopping threshold $\xi = 10^{-9}$) with different initial temperature parameters 0.001 and 1. The experimental results are listed in Figs. 2 and 3.

- For the different initial temperature parameters ($\tau_0 = 0.001$ and $\tau_0 = 1$), the updating rule Eq. (14) makes Opti-Softmax method converge to the same optimal temperature parameter $\tau_{\text{Opti}} = 0.0184$. The left sub-figures of Fig. 2(a) and (b) show that the updating curves increase gradually from 0.001 to 0.0184 with 240 iterations and decrease gradually from 1 to 0.0184 with 95 iterations, respectively. Opti-Softmax method can find the optimal temperature parameter without depending on the initial temperature parameter.
- Eq. (14) brings about the convergence of $(H_{\vec{z}} - H_{\vec{p}})^2$ (i.e., the information-loss term) and $H_{\vec{p}}^2$ (i.e., the diversity term), as shown in the right sub-figures of Fig. 2(a) and (b). Meanwhile, with the increase of temperature parameter, $(H_{\vec{z}} - H_{\vec{p}})^2$ decreases gradually (i.e., the information-loss decreases gradually), while $H_{\vec{p}}^2$ increases gradually (i.e., the diversity decreases gradually).
- The enhancement factor λ affects the number of iterations and the selection of optimal temperature parameter. Let λ change from 0 to 1.5 in step of 0.05. The number of iterations and optimal temperature parameters corresponding to the different initial temperature parameters $\tau_0 = 0.001$ and $\tau_0 = 1$ are summarized in Fig. 3. We can see that the numbers of iterations decrease firstly and then increase with the increase of enhancement factor, as shown in Fig. 3(a) and (b). In Fig. 3(c), the optimal temperature parameter decreases gradually with the increase of λ. When $\lambda > 1$, the number of iterations shows a trend of decrease. This provides us an enlightenment to select an appropriate enhancement factor, because the larger λ results in the smaller τ_{Opti} and further leads to the larger information-loss. Usually, we select the enhancement factor in the interval (0, 1] when the equivalent vector as shown in Eq. (6) is used.

[1] The source code of Opti-Softmax method has been uploaded on https://pan.baidu.com/s/1DgvnC23mtaIAiKc4KNOFpQ. The interested readers can use Opti-Softmax method to handle any other types of original vectors and validate the experimental results.

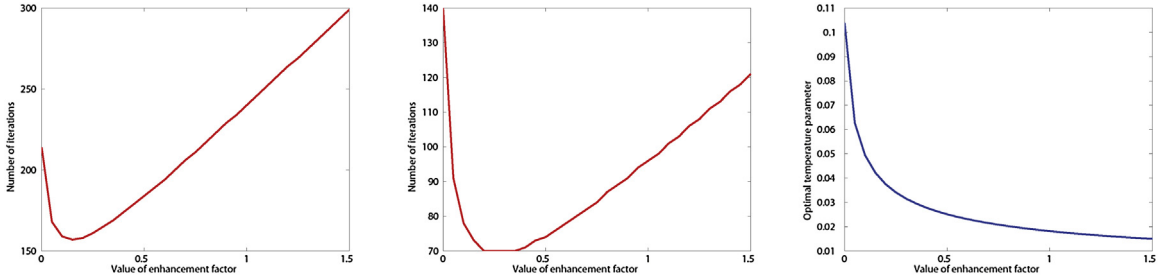


(a) $\tau_0 = 0.001$, $\lambda = 1$ and $\xi = 10^{-9}$



(b) $\tau_0 = 1$, $\lambda = 1$ and $\xi = 10^{-9}$

Fig. 2. The optimal temperature parameter τ_{Opti} is 0.0184 on the original vector $\vec{x} = (-7013.7933, -7282.7121, 649.9646, 4515.7862, -2025.9390, -2831.6297, -4294.4118, 7372.7049, 2528.2535, -5176.5538)$ and the corresponding probability vector $\vec{p}$ is (0.0017, 0.0015, 0.0403, 0.1976, 0.0134, 0.0096, 0.0053, 0.6394, 0.0873, 0.0037).



(a) $\tau_0 = 0.001$ and $\xi = 10^{-9}$ (b) $\tau_0 = 1$ and $\xi = 10^{-9}$ (c) The optimal temperature parameters

Fig. 3. The influence of enhancement factor λ on the number of iterations and the optimal temperature parameter, where the original vector $\vec{x}$ is (−7013.7933, − 7282.7121, 649.9646, 4515.7862, − 2025.9390, − 2831.6297, − 4294.4118, 7372.7049, 2528.2535, − 5176.5538).

In comparison to the exploration-only and exploitation-only methods, we validate the practical performance of Opti-Softmax method when dealing with $\mathcal{D}$-armed bandit problems ($\mathcal{D} =$ 5, 10, 20 and 50) in reinforcement learning. There are two types of bandit-arms in this experiment: the odd-numbered bandit-arm returns a reward 1 with probability 0.4 and the even-numbered bandit-arm returns a reward 1 with probability 0.2. The game-playing turns for $\mathcal{D} = 5, 10$ and $\mathcal{D} = 20, 50$ are 3000 and 5000, respectively. For each method, we repeat the game 100 times and record the average reward of gambler. The experimental results are presented in Fig. 4, where the parameters of Opti-Softmax method are set as $\tau_0 = 1$, $\lambda = 1$ and $\xi = 10^{-5}$. In Fig. 4, we can see that Opti-Softmax method obtains the significantly better performances than the exploration-only and exploitation-only methods. The average rewards of Opti-Softmax method approximates to 0.4 which is the highest reward that the gambler can obtain in the current

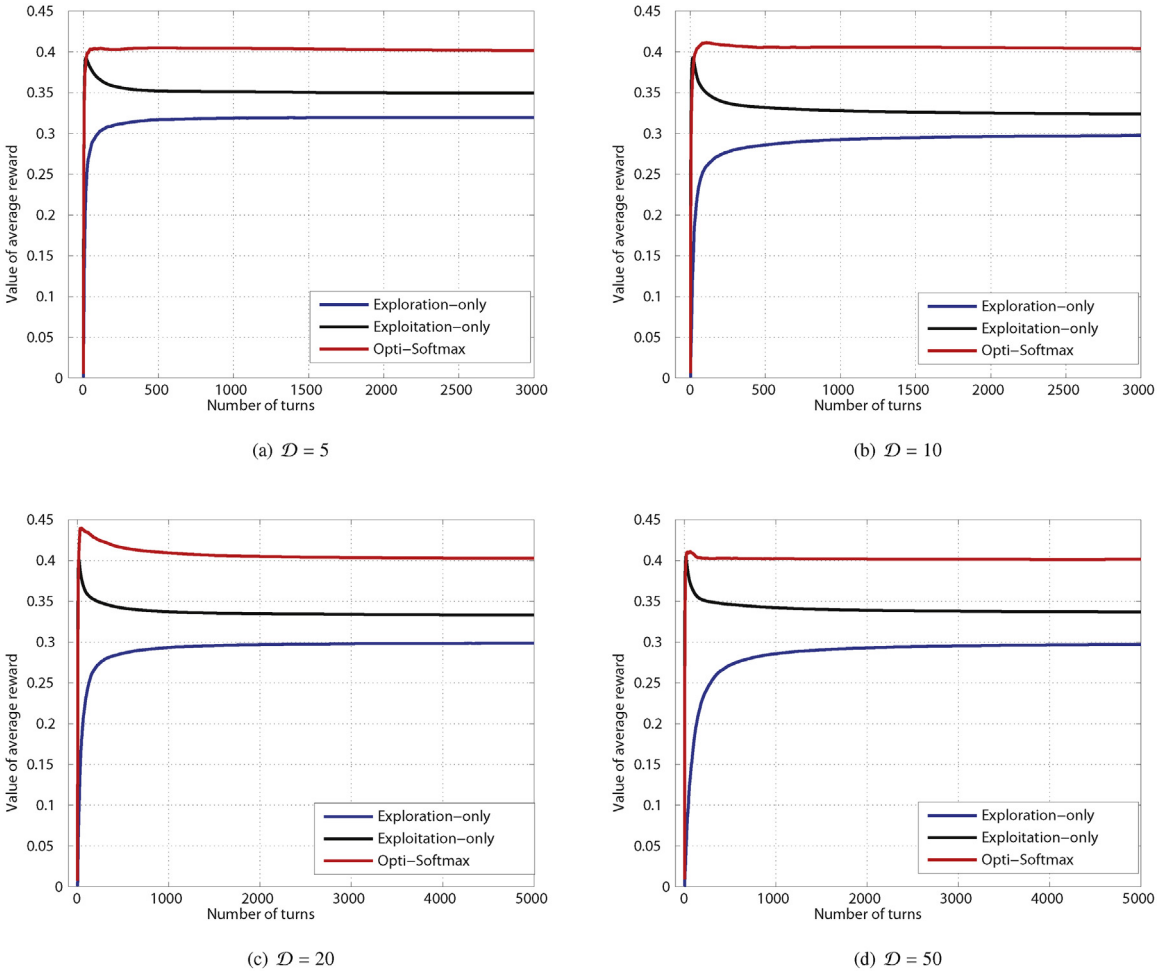


Fig. 4. The comparison of Exploration-only, Exploitation-only and Opti-Softmax ($\tau_0 = 1$, $\lambda = 1$ and $\xi = 10^{-5}$) when solving $\mathcal{D}$-armed bandit problems.

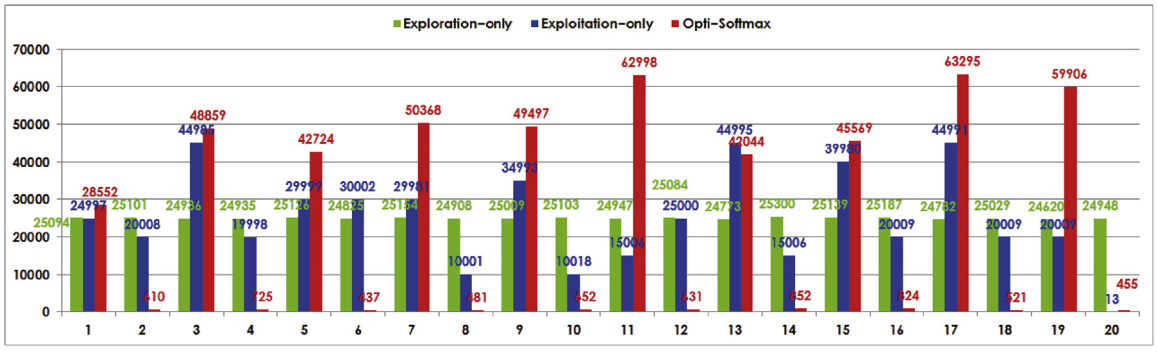


Fig. 5. The number that each bandit-arm in 20-armed bandit problem is selected. (For interpretation of the references to color in the text, the reader is referred to the web version of this article.)

experimental setting. Let $\mathcal{N}$ represent the number of game-playing turns. When $\mathcal{N} \to +\infty$, the average rewards Q of exploration-only, exploitation-only and Opti-Softmax method are calculated as follows:

$$\begin{cases} \text{Exploration} - \text{only}: & Q \to \frac{1}{2} \times 0.4 + \frac{1}{2} \times 0.2 = 0.3 \\ \text{Exploitation} - \text{only}: & Q \to \frac{0.4}{0.4+0.2} \times 0.4 + \frac{0.2}{0.4+0.2} \times 0.2 \approx 0.33 \\ \text{Opti} - \text{Softmax}: & Q \to 1 \times 0.4 + 0 \times 0.2 = 0.4 \end{cases} . \quad (15)$$

Eq. (15) reflects that the bandit-arms make the gambler obtain the reward with probability 0.4 are selected with probability 1 in Opti-Softmax method. In order to confirm this conclusion, we carry out a supplementary experiment as shown in Fig. 5. For 20-armed bandit problem, we calculate the numbers that each bandit-arm is selected in 5000×100 turns. We can easily find that the total number that the odd-numbered bandit-arms are selected by Opti-Softmax method (red bars) approximates to 493812, i.e., the probability that the odd-numbered bandit-arms are selected is $\frac{493812}{500000} = 0.9876 \approx 1$. For the exploration-only and exploitation-only methods, the probabilities that the odd-numbered bandit-arms are selected are $\frac{249580}{500000} = 0.4992 \approx \frac{1}{2}$ (green bars) and $\frac{329936}{500000} = 0.6599 \approx \frac{0.4}{0.4+0.2}$ (blue bars), respectively.

5. Conclusions and further works

By designing the efficient evaluation function and updating rule, this paper proposes a useful and simple method named Opti-Softmax to determine the optimal temperature parameter for Softmax function in reinforcement learning. The experimental results demonstrate that Opti-Softmax method is feasible and effective, which cannot only find the optimal temperature parameter for Softmax function with less iterations, but also make the gambler obtain the higher reward when playing $\mathcal{D}$-armed bandit games. In fact, Opti-Softmax method is an uncertainty reduction-based parameter optimization technology. In the future works, we will study the integration of Opti-Softmax method with uncertainty reduction-based machine learning methods, e.g., random weight networks [7,9,10], representation learning [3,20], incomplete information handling [2,14].

Acknowledgments

We thank the editors and three anonymous reviewers whose valuable comments and suggestions help us to improve this paper significantly after two rounds of review. This paper was supported by National Natural Science Foundations of China (61503252 and 61473194), China Postdoctoral Science Foundation (2016T90799), Scientific Research Foundation of Shenzhen University for Newly-introduced Teachers (2018060) and Shenzhen-Hong Kong Technology Cooperation Foundation (SGLH20161209101100926).

References

[1] B. Abdulhai, R. Pringle, G. Karakoulas, Reinforcement learning for true adaptive traffic signal control, J. Transp. Eng. 129 (3) (2003) 278–285.
[2] R. Ashfaq, X. Wang, J. Huang, H. Abbas, Y. He, Fuzziness based semi-supervised learning approach for intrusion detection system, Inf. Sci. 378 (2017) 484–497.
[3] Y. Bengio, A. Courville, P. Vincent, Representation learning: a review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (8) (2013) 1798–1828.
[4] C.M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.
[5] D. Bohning, Multinomial logistic regression algorithm, Ann. Inst. Stat. Math. 44 (1) (1992) 197–200.
[6] S. Branavan, H. Chen, L. Zettlemoyer, R. Barzilay, Reinforcement learning for mapping instructions to actions, Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing (2009) 82–90.
[7] W.P. Cao, X.Z. Wang, Z. Ming, J.Z. Gao, A review on neural networks with random weights, Neurocomputing 275 (2018) 278–287.
[8] A. Garivier, E. Moulines, On upper-confidence bound policies for switching bandit problems, Proceedings of 2011 International Conference on Algorithmic Learning Theory (2011) 174–188.
[9] Y. He, X. Wang, J. Huang, Fuzzy nonlinear regression analysis using a random weight network, Inf. Sci. 364–365 (2016) 222–240.
[10] Y. He, C. Wei, H. Long, R. Ashfaq, J. Huang, Random weight network-based fuzzy nonlinear regression for trapezoidal fuzzy number data, Appl. Soft Comput. (2017), http://dx.doi.org/10.1016/j.asoc.2017.08.006.
[11] Y. Kohno, T. Takahashi, Loosely symmetric reasoning to cope with the speed-accuracy trade-off, Proceedings of 2012 Joint 6th International Conference on Soft Computing and Intelligent Systems and 13th International Symposium on Advanced Intelligent Systems (2012) 1166–1171.
[12] D. Koulouriotis, A. Xanthopoulos, Reinforcement learning and evolutionary algorithms for non-stationary multi-armed bandit problems, Appl. Math. Comput. 196 (2) (2008) 913–922.
[13] V. Kuleshov, D. Precup, Algorithms for Multi-Armed Bandit Problems, 2014 arXiv:1402.6028.
[14] Y. Lan, R. Zhao, W. Tang, An inspection-based price rebate and effort contract model with incomplete information, Comput. Ind. Eng. 83 (2015) 264–272.
[15] J. Masci, U. Meier, D. Ciresan, J. Schmidhuber, Stacked convolutional auto-encoders for hierarchical feature extraction, Lecture Notes Comput. Sci. 6791 (2011) 52–59.
[16] L. Paletta, A. A. Pinz, Active object recognition by view integration and reinforcement learning, Robot. Auton. Syst. 31 (1) (2000) 71–86.
[17] R. Sutton, A. Barto, Reinforcement Learning: An Introduction, The MIT Press, Cambridge, MA, 1998.
[18] A. Sykulski, N. Adams, N.R. Jennings, On-line adaptation of exploration in the one-Armed bandit with covariates problem, Proceedings of 2010 Ninth International Conference on Machine Learning and Applications (2010) 459–464.
[19] M. Tokic, G. Palm, Value-difference based exploration: adaptive control between epsilon-greedy and softmax, Lecture Notes Artif. Intell. 7006 (2011) 335–346.
[20] M. Yang, P. Zhu, F. Liu, L. Shen, Joint representation and pattern learning for robust face recognition, Neurocomputing 168 (2015) 70–80.
[21] Z. Zhou, Machine Learning, Tsinghua University Press, 2016.