[Applied Soft Computing 70 (2018) 80–85](https://doi.org/10.1016/j.asoc.2018.05.012)

Contents lists available at [ScienceDirect](http://www.sciencedirect.com/science/journal/15684946)

Applied Soft Computing

journal homepage: [www.elsevier.com/locate/asoc](http://www.elsevier.com/locate/asoc)

Determining the optimal temperature parameter for Softmax function in reinforcement learning

Yu-Lin He[a*,*](#_bookmark0)[b](#_bookmark1)[*,*](#_bookmark0)[∗](#_bookmark2)[,](#_bookmark1) Xiao-Liang Zhang[a*,*](#_bookmark0)[b](#_bookmark1)[,](#_bookmark0) Wei Ao[a](#_bookmark0)*[,](#_bookmark0)*[b](#_bookmark1)[,](#_bookmark0) Joshua Zhexue Huang[a*,*](#_bookmark0)[b](#_bookmark1)[*,*](#_bookmark0)[∗](#_bookmark2)

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*

## a r t i c l e i n f o a b s t r a c t

*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

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 *D*-armed bandit problems demonstrate the feasibility and effectiveness of Opti-Softmax method.

© 2018 Elsevier B.V. All rights reserved.

#### Introduction

Softmax function is a normalized exponential function [[4]](#_bookmark21) which transforms a -dimensional original vector with arbitrary real val- ues into a -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],](#_bookmark22) artiﬁcial neural networks [[15],](#_bookmark26) reinforcement learning [[17].](#_bookmark27) In gen- eral, 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 func- tion with temperature parameter [[17]](#_bookmark27) is used to convert the action rewards into the action probabilities in reinforcement learning.

*D*

*D*

The temperature parameter is an important learning param- eter 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 proba- bilities 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

∗ 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](mailto:yulinhe@szu.edu.com)

(Y.-L. He), [zhangxlas@163.com](mailto:zhangxlas@163.com) (X.-L. Zhang), [aowei2016@email.szu.edu.cn](mailto:aowei2016@email.szu.edu.cn) (W. Ao), [zx.huang@szu.edu.cn](mailto:zx.huang@szu.edu.cn) (J.Z. Huang).

-armed bandit [[21]](#_bookmark30) which is a classical action selection problem of reinforcement learning. Some representative studies related to Softmax action selection are summarized as follows. Koulourio- tis and Xanthopoulos in [[12]](#_bookmark23) examined Softmax algorithm with temperature parameter 0.3. Tokic and Palm in [[19]](#_bookmark28) tested the performances of Softmax action selection algorithms with tem- perature parameters 0.04, 0.1, 1, 10, 25 and 100. In [[13],](#_bookmark25) 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].](#_bookmark29) 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

*D*

-armed bandit problem.

*D*

The simply Softmax function will be a very efﬁcient action selec- tion strategy to solve -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 - 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

*D*

*D*

<https://doi.org/10.1016/j.asoc.2018.05.012>

1568-4946/© 2018 Elsevier B.V. All rights reserved.

*Y.-L. He et al. / Applied Soft Computing 70 (2018) 80–85* 81

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 *D*-

is the amount of information about the probability vector p˙ =

(*p*1*, p*2*, . . ., pD*),

.*z* Σ

Σ

*ı*

# 

exp

*zk ı*

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](#_bookmark3) states the problem formulations of Softmax function. Section [3](#_bookmark6) gives the Opti-

*pd* = Σ

exp *d*

# .

Σ∈ *p* = *hd d*

*d* 1

*D*

*k*=1

*,* (5)

Softmax method to determine the optimal temperature parameter

for Softmax function. Some experimental simulations are given in

Section [4.](#_bookmark14) Finally, Section [5](#_bookmark21) presents a brief conclusion to this paper.

#### Problem formulations of Softmax function

Given a *D*-dimensional original vector x˙ = (*x*1*, x*2*, . . ., xD*), *xd* ∈

*p* (0, 1), *D* 1; and > 0 is the enhancement factor.

The ﬁrst term in Eq. [(2)](#_bookmark7) is to measure the information-loss after

=

transforming the original vector x˙ into the probability vector p˙. Because *xd* ∈ R, *d* = 1*,* 2*, . . ., D*, we cannot obtain the information- amount of x˙ directly. Hence, a linear transformation is performed on the original vector x˙ and then generates its equivalent vector z˙ as

R, *d* = 1*,* 2*, . . ., D* and there exists *k* ∈ {1*,* 2*, . . ., D*} such that

*d*

*D*

[*x*

+ 2|*x*

| + 0*.*01]

) with the following Softmax function:

*xk* =*/* 0, it can be transformed into a *D*-dimensional probability vec-

*z* = Σ *xd* + 2|*x*min| + 0*.*01

*k*=1

*k*

min

*,* (6)

exp . *xd* Σ

tor p˙ = (*p , p , . . ., p*

1

2

*D*

*ı*

*ı*

where *x*min = min{*x*1*, x*2*, . . ., xD*}. For the original vector

∃ ∈ { *D*} =

x˙ =

*pd* = Σ

Σ∈*d*

*D*

*k*=1

exp . *xk*

*,* (1)

(*x*1*, x*2*, . . ., xD*), *k* 1*,* 2*, . . .,* , *xk /* 0, Eq. [(6)](#_bookmark4) ensures 0 < *zd* < 1

for *d* 1*,* 2*, . . .,* . The role of Eq. [(6)](#_bookmark4) is to facilitate the

∀ ∈ { *D*}

calculation of information-amount. We cannot calculate the

where *p* (0, 1), *D*

Σ

*d*=1

*pd* = 1 and *ı* > 0 is the temperature param-

information-amount of original vector directly if the elements of

original vector are beyond the interval (0, 1). Thus, we need to

eter which has an important inﬂuence on the transformation

performance of Softmax function. When *ı* + , *p* 1 , i.e., the

→ ∞ →*d*

diversity among *pd*s is small; when *ı* → 0, |*pi* − *pj*| |*xi* −*Dxj*|, *i, j* ∈ 1*,* 2*, . . .,* , *i / j*. i.e., the diversity among *pd*s is large. [Fig. 1](#_bookmark11) pro- vides an example to show the inﬂuence of *ı* 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 probabil- ity vector elements corresponding to *ı* = 1 (red bars) are almost

1

{ *D*} =

∃

20 0*.*05, while *ı* = 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 *D*-armed bandit, where each bandit-arm

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)](#_bookmark7) is to control the diversity among probability vector elements *p*1*, p*2*, . . ., pD*. Hp˙ attains its maximum In(*D*) at *p*1 = *p*2 = · · · = *pD* =

1 . We hope that the optimal temperature parameter *ı*Opti not only m*D* inimizes the information-loss of transformation but also maxi-

mizes the diversity among probability vector elements. Thus, we can get the optimality expression of *ı*Opti as

*ı*Opti = *arg* min*L*(*ı*) = *arg* min Σ(Hz˙ − Hp˙)2 + *h*H2Σ *.* (7)

*ı>*0

*ı>*0

p˙

provides a random reward for gambler. Assume x˙ = (*x*1*, x*2*, . . ., xD*)

is the reward vector corresponding to *D* bandit-arms. Then, *pd* is Let *E* = Σ*D* exp . *zd* Σ and *F* = Σ*D* Σ *zd* exp . *zd* ΣΣ, Hp˙ in Eq.

the probability with which the *d*-th bandit-arm is selected. *ı* → +∞

*d*=1

*ı*

*d*=1

*ı*

*ı*

[(4)](#_bookmark10) can be equivalently written as

will lead to the exploration-only state (the bandit-arms have almost

*ı*

*ı*

the same probabilities to be selected), while *ı* → 0 will result in the

are more easily selected). The key of solving *D*-armed bandit prob-

Σ*D* Σ exp . *zd* Σ

*ı*

Σ

*d*=1

*D*

*k*=1

exp

*zk*

*ı*

Σ exp . *zd* Σ ΣΣ

Σ

.

*zd*

ΣΣ

.

*zk*

*D*

*k*=1

exp

*zk*

*ı*

exploitation-only state (the bandit-arms with the higher rewards

lem is how to select the bandit-arms so that the gambler can obtain

Hp˙ = −

Σ . Σ In Σ . Σ

the maximal reward.

= −

*d*=1

Σ

*D*

*k*=1

exp . *zk* Σ

Σ exp . *zd* Σ

*d*

In exp

*ı*

*ı*

− In

*k*=1

exp

*ı*

*.*(8)

#### Determination of the optimal temperature parameter

**for Softmax function**

1 Σ*D* Σ

Σ*D*

ΣΣ*D*

ΣΣΣΣ

exp

*d*

*ı*

*ı* − In(*E*)

= − *E*

*F*

*k*=1

.*z* ΣΣ*z* ΣΣ

This section presents a new method named Opti-Softmax to

determine the optimal temperature parameter *ı*Opti for Softmax function. The following evaluation function *L*(*ı*) is ﬁrstly designed to measure the effectiveness of temperature parameter:

= − *E* + In(*E*)

Bringing Eq. [(8)](#_bookmark5) into Eq. [(2),](#_bookmark7) we can obtain

Σ *F* Σ2 Σ *F* Σ2

p˙

*L*(*ı*) =

Hz˙ + *E* − In(*E*)

+ *h*

*E* − In(*E*)

*.* (9)

*L*(*ı*) = (Hz˙ − Hp˙)2 + *h*H2*,* (2)

where

It is very difﬁcult to determine the analytic formulation of *ı*Opti

*dL*(*ı*)

=

*zd*

H Σ*D*

z˙ = −

*zd*

*d*=1

In(

) (3)

by solving *dı*

to *ı*

, we try to ﬁnd the optimal Eq. [(9)](#_bookmark8) by calculating

0. Because *E* and *F* are the functions with respect

*E* or *F* which can minimize *L*(*T*) in

is the amount of information about z˙ = (*z*1*, z*2*, . . ., zD*) which is the

Σ

*D*

*d*=1

# 

equivalent vector of x˙, *zd* ∈ (0, 1),

*zd* = 1;

*dL*(*ı*)

*dE*

= 2 ΣH

*F* In(*E*)Σ. *F*

1 Σ + 2*h* Σ *F* − In(*E*)Σ. *F* 1 Σ

Σ*D* = −2 . *F* + *E* ΣΣH + (1 + *h*) *F* − (1 + *h*) In(*E*)Σ

z˙ + *E* −

− *E*2 − *E*

*E*

− *E*2 − *E*

H

p˙ = −

*pd*In(*pd*) (4)

*E*2

z˙

*E*

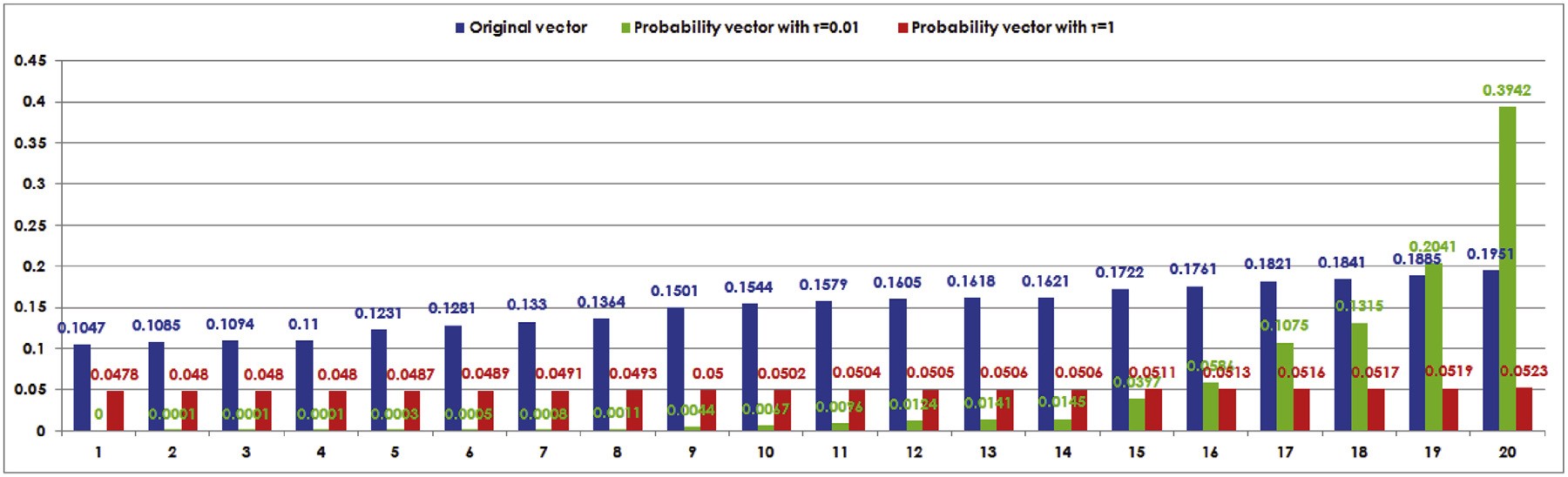
# 

*d*=1

= 0

(10)

82 *Y.-L. He et al. / Applied Soft Computing 70 (2018) 80–85*



**Fig. 1.** The inﬂuence of *ı* 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 and the second experiment is to use Opti-Softmax method to deal

+

*E* − In(*E*)

*dL*(*ı*) 2 Σ *F*

*dF* = *E*

Hz˙ + *E* − In(*E*)

In the ﬁrst experiment, a 10-dimensional original vector

Σ 2*h* Σ *F* Σ

# 

*E*

with the *D*-armed bandit problems in reinforcement learning. [1](#_bookmark15)

2 Σ Σ

Hz˙

*F*

+ (1 + *h*) *E* − (1 + *h*)In(*E*)

*.* (11)

### 

=

*E*

→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

= 0

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

working performances of Opti-Softmax method (the enhancement

factor *h* = 1 and the stopping threshold *‡* = 10−9) with different ini- tial temperature parameters 0.001 and 1. The experimental results

Hz˙

*F*

+ (1 + *h*) *E* − (1 + *h*)In(*E*) = 0*,* (12)

are listed in [Figs. 2 and 3.](#_bookmark16)

i.e.,

Σ*D* Σ

*zd* exp

*ı*

*d*=1

# . ΣΣ

ΣΣ*D*

=

*d*=1

# . ΣΣΣ

ΣΣ*D*

*d*=1

# . ΣΣ Σ

−

* For the different initial temperature parameters (*ı*0 = 0.001

exp

*zd*

*ı*

Hz˙ 1 + *h*

*.* (13)

and *ı*0 = 1), the updating rule Eq. [(14)](#_bookmark13) makes Opti-Softmax method converge to the same optimal temperature parameter *ı*Opti = 0.0184. The left sub-ﬁgures of [Fig. 2](#_bookmark16)(a) and (b) show that

Eq. [(13)](#_bookmark12) can be further simpliﬁed as

. ΣΣ*z*exp

*zd*

*ı*

exp

*zd*

*ı*

In

Σ

the updating curves increase gradually from 0.001 to 0.0184 with

*D*

Σ

*d*=1

# 

exp

*d*

*ı*

*zd d*

In

*d*=1

−

z˙

+*h*

*ı*

# 

exp

*d*

*ı*

1

240 iterations and decrease gradually from 1 to 0.0184 with 95 iterations, respectively. Opti-Softmax method can ﬁnd the opti-

*ı* = [ΣΣ](#_bookmark13)*D*

*d*=1

. *z* ΣΣΣ

ΣΣ*D*

. *z* ΣΣ

H Σ *.* (14)

Eq. [(14)](#_bookmark13) is the heuristic updating rule of Opti-Softmax method to determine the optimal temperature parameter *ı*Opti. According to this updating rule, Opti-Softmax method as shown in Algorithm 1 gives an iterative procedure to determine *ı*Opti.

mal temperature parameter without depending on the initial temperature parameter.

**Algorithm 1.** Opti-Softmax method

1: **Input:** The original vector x˙ = (*x*1*, x*2*, . . ., xD*), *xd* ∈ R, *d* = 1*,* 2*, . . ., D* and *k* ∈ {1*,* 2*, . . ., D*}, *xk* =*/* 0; the enhancement factor *h* > 0; the stopping threshold *‡* > 0; the initial temperature parameter *ı*0 > 0.

∃

2: **Output:** The probability vector p˙ = (*p*1*, p*2*, . . ., pD*), *pd* ∈ (0, 1),

Σ*D*

*d* = 1*,* 2*, . . ., D*, *pd* = 1 and the optimal temperature parameter

*d*=1

*ı*

Opti

.

* Eq. [(14)](#_bookmark13) brings about the convergence of (Hz˙ − Hp˙)2 (i.e., the information-loss term) and H2 (i.e., the diversity term), as shown

p˙

in the right sub-ﬁgures of [Fig. 2](#_bookmark16)(a) and (b). Meanwhile, with the

increase of temperature parameter, (Hz˙ − Hp˙)2 decreases grad-

ually (i.e., the information-loss decreases gradually), while H2 increases gradually (i.e., the diversity decreases gradually). p˙

* The enhancement factor *h* affects the number of iterations and the

selection of optimal temperature parameter. Let *h* 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 *ı*0 = 0.001 and *ı*0 = 1 are summarized in

3: Calculating the equivalent vector z˙ = (*z*1*, z*2*, . . ., zD*), *zd* ∈ (0, 1),

Σ*D*

*d* = 1*,* 2*, . . ., D*, 1 *zd* = 1 according to Eq. [(6);](#_bookmark4)

4:

*d*=

Calculating the information-amount Hz˙ of z according to Eq. [(3);](#_bookmark9)

˙

5: **repeat**

6: *ı*Opti = *ı*0;

Σ Σ . ΣΣ

*D z* exp *zd*

*d*=1

*d*

*ı*Opti

ΣΣ . ΣΣΣ ΣΣ . ΣΣ Σ

7: *ı*0 = *D zd D zd* Hz˙ ; 8: **until** .*ı* − *ı* . *< ‡*

. Σ*z*exp

exp

*d*=1

*ı*Opti

In

exp

*d*=1

*ı*Opti

− 1+*h*

Opti

9: *ı*Opti = *ı*0;

0

*d ı*Opti

[Fig. 3.](#_bookmark17) We can see that the numbers of iterations decrease ﬁrstly and then increase with the increase of enhancement factor, as shown in [Fig. 3](#_bookmark17)(a) and (b). In [Fig. 3](#_bookmark17)(c), the optimal temperature parameter decreases gradually with the increase of *h*. When *h* > 1, the number of iterations shows a trend of decrease. This pro- vides us an enlightenment to select an appropriate enhancement factor, because the larger *h* results in the smaller *ı*Opti and fur-

ther 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)](#_bookmark4) is used.

10: *pd* = Σ*D* . *zk* Σ , *d* = 1*,* 2*, . . ., D*;

*ı*Opti

exp

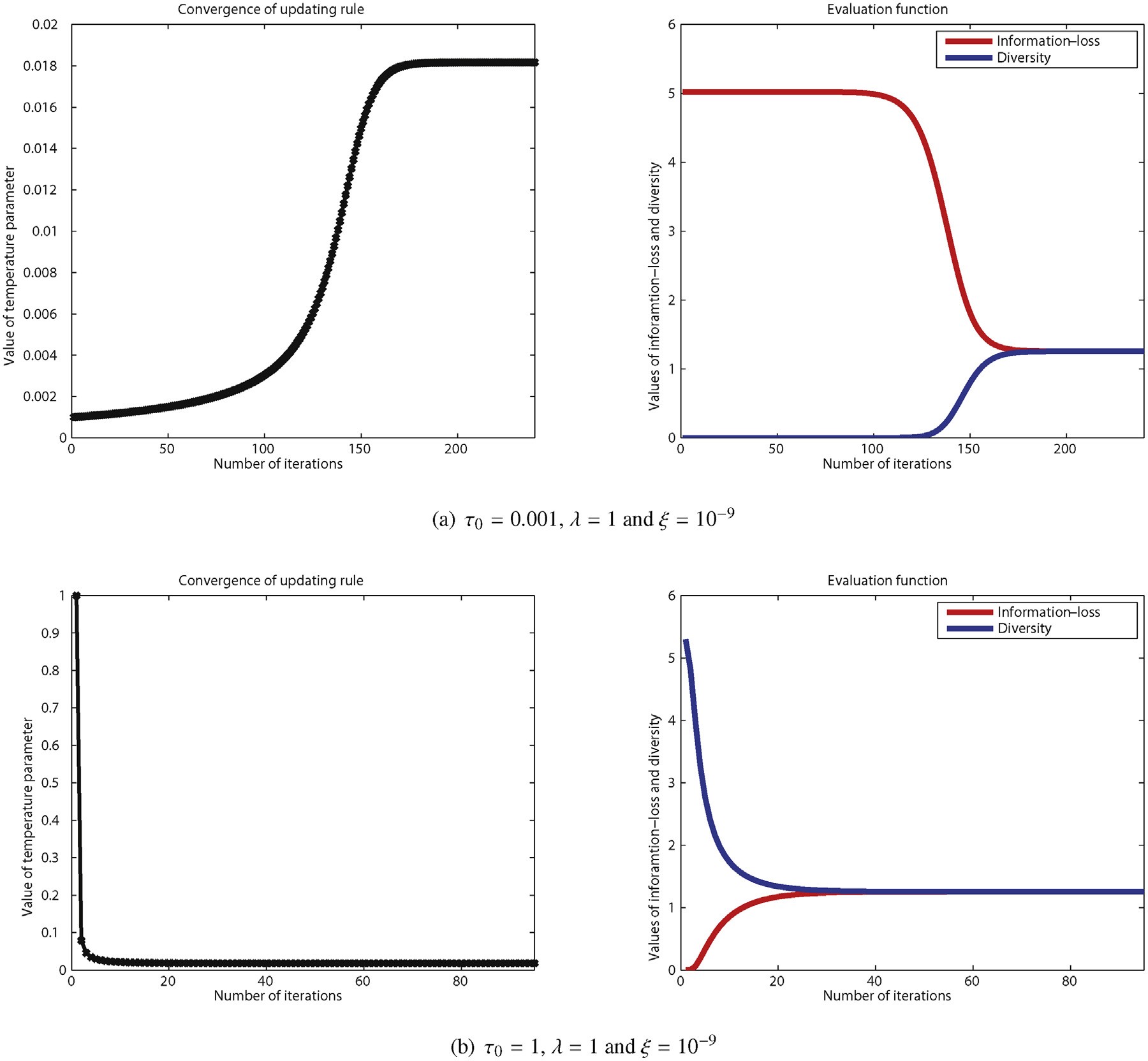
*k*=1

#### 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)](#_bookmark13) is convergent

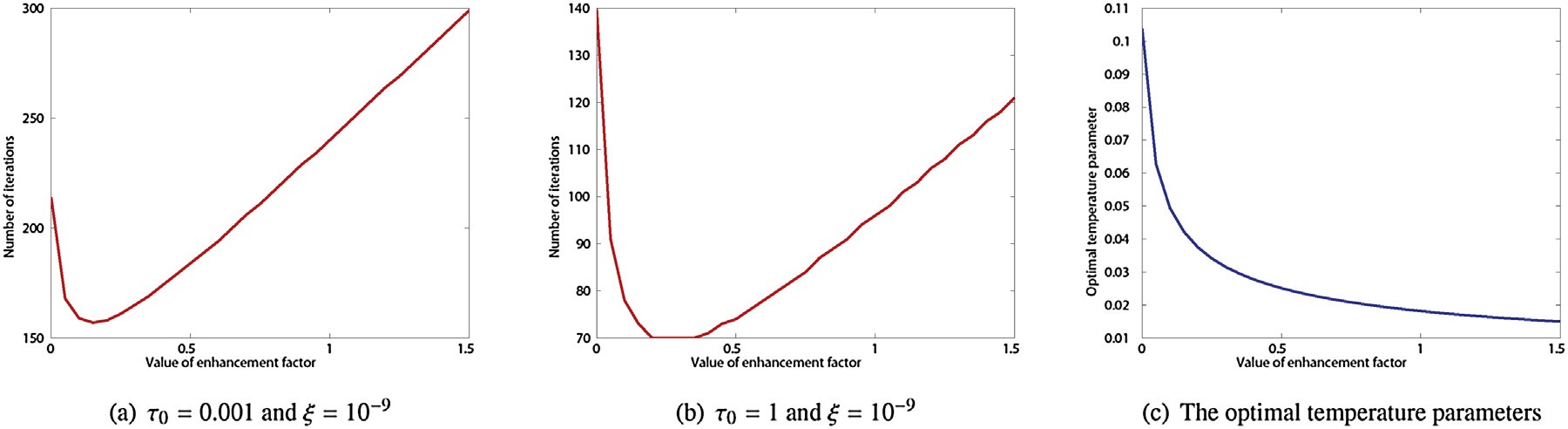
1 The source code of Opti-Softmax method has been uploaded on [https://pan.](https://pan.baidu.com/s/1DgvnC23mtaIAiKc4KNOFpQ) [baidu.com/s/1DgvnC23mtaIAiKc4KNOFpQ](https://pan.baidu.com/s/1DgvnC23mtaIAiKc4KNOFpQ). The interested readers can use Opti- Softmax method to handle any other types of original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

*Y.-L. He et al. / Applied Soft Computing 70 (2018) 80–85* 83



**Fig. 2.** The optimal temperature parameter *ı*Opti is 0.0184 on the original vector →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 p˙ is (0.0017, 0.0015, 0.0403, 0.1976, 0.0134, 0.0096, 0.0053, 0.6394, 0.0873, 0.0037).



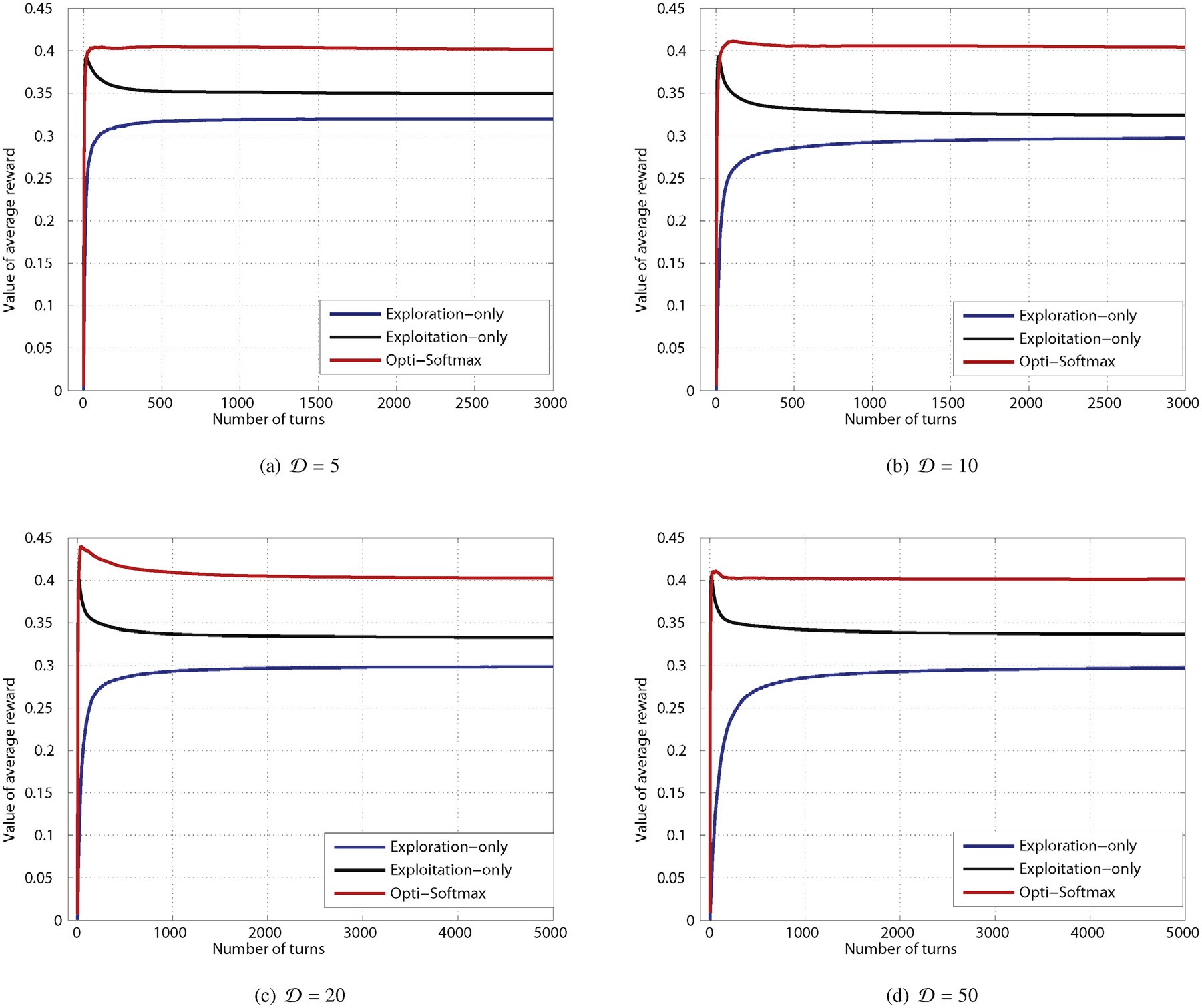
**Fig. 3.** The inﬂuence of enhancement factor *h* on the number of iterations and the optimal temperature parameter, where the original vector x˙ is (−7013.7933, − 7282.7121, 649.9646, 4515.7862, − 2025.9390, − 2831.6297, − 4294.4118, 7372.7049, 2528.2535, − 5176.5538).

In comparison to the exploration-only and exploitation-only methods, we validate the practical performance of Opti-Softmax method when dealing with -armed bandit problems ( 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 *D* = 5*,* 10 and *D* = 20*,* 50 are 3000 and 5000,

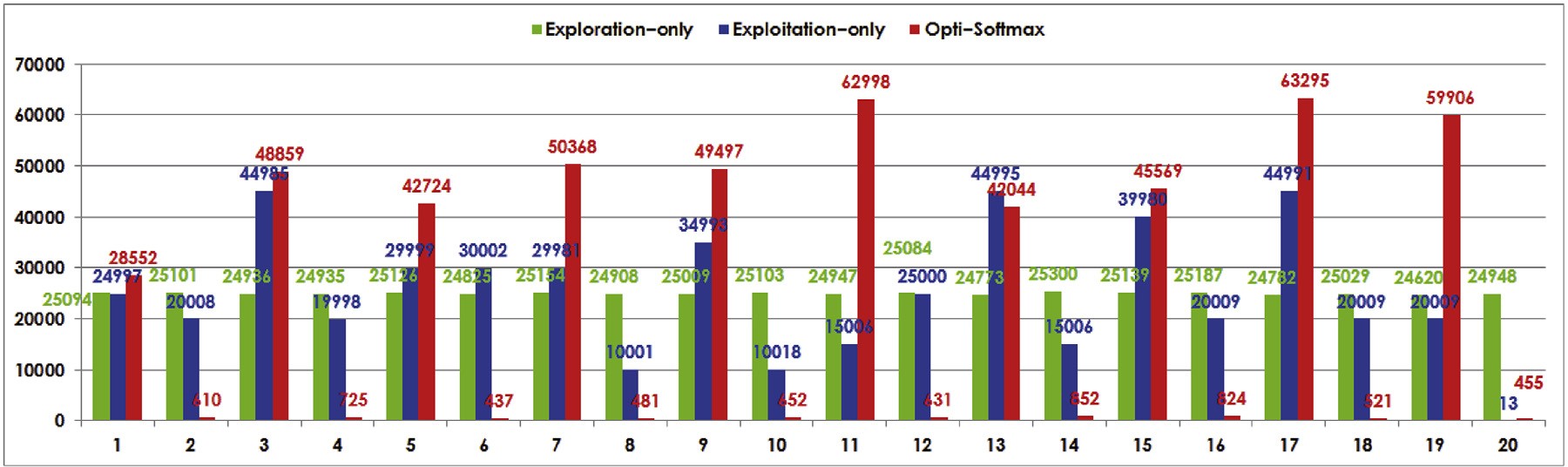
*D D* =

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,](#_bookmark18) where the parameters of Opti-Softmax method are set as *ı*0 = 1, *h* = 1 and *‡* = 10−5. In [Fig. 4,](#_bookmark18) 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

84 *Y.-L. He et al. / Applied Soft Computing 70 (2018) 80–85*



**Fig. 4.** The comparison of Exploration-only, Exploitation-only and Opti-Softmax (*ı*0 = 1, *h* = 1 and *‡* = 10−5) when solving *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 represent the number of game-playing turns. When , the average rewards *Q* of exploration-only, exploitation-only and Opti-Softmax method are calculated as fol- lows:

*N* → +∞

*N*

Exploration only : *Q* 1 0*.*4 1 0*.*2 0*.*3

→ × + ×

2

2

⎨ (15)

×

Exploitation − only : *Q* → 0*.*4 × 0*.*4 + 0*.*2 × 0*.*2 ≈ 0*.*33 *.*

⎩⎪

⎧⎪ − =

sion, we carry out a supplementary experiment as shown in [Fig. 5.](#_bookmark19) For 20-armed bandit problem, we calculate the num- bers 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-

500000

0*.*4 + 0*.*2

Opti − Softmax : *Q* → 1 × 0*.*4 + 0 × 0*.*2 = 0*.*4

Eq. [(15)](#_bookmark20) reﬂects that the bandit-arms make the gambler obtain

0*.*4 + 0*.*2

the exploration-only and exploitation-only methods, the probabil-

500000

numbered bandit-arms are selected is 493812 = 0*.*9876 ≈ 1. For

ities that the odd-numbered bandit-arms are selected are 249580 =

0*.*4992 ≈ 1 (green bars) and 329936 = 0*.*6599 ≈ 0*.*4 (blue bars),

the reward with probability 0.4 are selected with probability

1 in Opti-Softmax method. In order to conﬁrm this conclu-

2

respectively.

500000

0*.*4+0*.*2

*Y.-L. He et al. / Applied Soft Computing 70 (2018) 80–85* 85

#### 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 experimen- tal results demonstrate that Opti-Softmax method is feasible and effective, which cannot only ﬁnd the optimal temperature param- eter for Softmax function with less iterations, but also make the gambler obtain the higher reward when playing -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],](#_bookmark24) representation learning [[3,20],](#_bookmark32) incomplete infor- mation handling [[2,14].](#_bookmark31)

*D*

#### 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](http://refhub.elsevier.com/S1568-4946(18)30275-8/sbref0005) [adaptive trafﬁc signal control, J. Transp. Eng. 129 (3) (2003) 278–285.](http://refhub.elsevier.com/S1568-4946(18)30275-8/sbref0005)
  2. [R. Ashfaq, X. Wang, J. Huang, H. Abbas, Y. He, Fuzziness based](http://refhub.elsevier.com/S1568-4946(18)30275-8/sbref0010)

[semi-supervised learning approach for intrusion detection system, Inf. Sci.](http://refhub.elsevier.com/S1568-4946(18)30275-8/sbref0010) [378 (2017) 484–497.](http://refhub.elsevier.com/S1568-4946(18)30275-8/sbref0010)

* 1. [Y. Bengio, A. Courville, P. Vincent, Representation learning: a review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (8) (2013) 1798–1828.](http://refhub.elsevier.com/S1568-4946(18)30275-8/sbref0015)
  2. [C.M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.](http://refhub.elsevier.com/S1568-4946(18)30275-8/sbref0020)
  3. [D. Bohning, Multinomial logistic regression algorithm, Ann. Inst. Stat. Math. 44 (1) (1992) 197–200.](http://refhub.elsevier.com/S1568-4946(18)30275-8/sbref0025)
  4. [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.](http://refhub.elsevier.com/S1568-4946(18)30275-8/sbref0030)
  5. [W.P. Cao, X.Z. Wang, Z. Ming, J.Z. Gao, A review on neural networks with random weights, Neurocomputing 275 (2018) 278–287.](http://refhub.elsevier.com/S1568-4946(18)30275-8/sbref0035)
  6. [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.](http://refhub.elsevier.com/S1568-4946(18)30275-8/sbref0040)
  7. [Y. He, X. Wang, J. Huang, Fuzzy nonlinear regression analysis using a random weight network, Inf. Sci. 364–365 (2016) 222–240.](http://refhub.elsevier.com/S1568-4946(18)30275-8/sbref0045)
  8. 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.](http://dx.doi.org/10.1016/j.asoc.2017.08.006)
  9. [Y. Kohno, T. Takahashi, Loosely symmetric reasoning to cope with the speed-accuracy trade-off, Proceedings of 2012 Joint 6th International](http://refhub.elsevier.com/S1568-4946(18)30275-8/sbref0055)

[Conference on Soft Computing and Intelligent Systems and 13th International](http://refhub.elsevier.com/S1568-4946(18)30275-8/sbref0055) [Symposium on Advanced Intelligent Systems (2012) 1166–1171.](http://refhub.elsevier.com/S1568-4946(18)30275-8/sbref0055)

* 1. [D. Koulouriotis, A. Xanthopoulos, Reinforcement learning and evolutionary algorithms for non-stationary multi-armed bandit problems, Appl. Math. Comput. 196 (2) (2008) 913–922.](http://refhub.elsevier.com/S1568-4946(18)30275-8/sbref0060)
  2. [V. Kuleshov, D. Precup, Algorithms for Multi-Armed Bandit Problems, 2014 arXiv:1402.6028.](http://refhub.elsevier.com/S1568-4946(18)30275-8/sbref0065)
  3. [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.](http://refhub.elsevier.com/S1568-4946(18)30275-8/sbref0070)
  4. [J. Masci, U. Meier, D. Ciresan, J. Schmidhuber, Stacked convolutional](http://refhub.elsevier.com/S1568-4946(18)30275-8/sbref0075)

[auto-encoders for hierarchical feature extraction, Lecture Notes Comput. Sci.](http://refhub.elsevier.com/S1568-4946(18)30275-8/sbref0075) [6791 (2011) 52–59.](http://refhub.elsevier.com/S1568-4946(18)30275-8/sbref0075)

* 1. [L. Paletta, A. A. Pinz, Active object recognition by view integration and reinforcement learning, Robot. Auton. Syst. 31 (1) (2000) 71–86.](http://refhub.elsevier.com/S1568-4946(18)30275-8/sbref0080)
  2. [R. Sutton, A. Barto, Reinforcement Learning: An Introduction, The MIT Press, Cambridge, MA, 1998.](http://refhub.elsevier.com/S1568-4946(18)30275-8/sbref0085)
  3. [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.](http://refhub.elsevier.com/S1568-4946(18)30275-8/sbref0090)
  4. [M. Tokic, G. Palm, Value-difference based exploration: adaptive control between epsilon-greedy and softmax, Lecture Notes Artif. Intell. 7006 (2011) 335–346.](http://refhub.elsevier.com/S1568-4946(18)30275-8/sbref0095)
  5. [M. Yang, P. Zhu, F. Liu, L. Shen, Joint representation and pattern learning for robust face recognition, Neurocomputing 168 (2015) 70–80.](http://refhub.elsevier.com/S1568-4946(18)30275-8/sbref0100)
  6. [Z. Zhou, Machine Learning, Tsinghua University Press, 2016.](http://refhub.elsevier.com/S1568-4946(18)30275-8/sbref0105)