

# A numerical analysis of learning coefficient in radial basis function network

SATORU TOKUDA<sup>1,a)</sup> KENJI NAGATA<sup>1,b)</sup> MASATO OKADA<sup>1,c)</sup>

**Abstract:** The radial basis function (RBF) network is a regression model that use the sum of radial basis functions such as Gaussian functions. It has recently been widely applied to spectral deconvolution such as X-ray photoelectron spectroscopy data analysis, which enables us to estimate the electronic state of matter from the spectral peak positions. For models with a hierarchy such as the RBF network, Bayesian learning provides better generalization performance than the maximum likelihood estimation. In Bayesian learning, the learning coefficient is well-known as the coefficients of the leading terms for the asymptotic expansion of generalization error and stochastic complexity. However, these coefficients have not been clarified in most models. We propose here a novel method for calculating the learning coefficient by using the exchange Monte Carlo method. In addition, we calculated the learning coefficient in the RBF networks and verified the efficiency of the proposed method by comparing theoretical and experimental values.

**Keywords:** Radial basis function network, Learning coefficient, Exchange Monte Carlo method

## 1. Introduction

The radial basis function (RBF) network is an artificial neural network that is used in function approximation, time series prediction, and system control. This network has been widely applied recently in the field of condensed matter physics and chemistry for spectral deconvolution such as X-ray photoelectron spectroscopy data analysis, which makes it possible to estimate the electronic state of matter from the positions of the spectral peaks [1]. In the field of geoscience, it was reported that the estimation of the number of bases and the parameters of each basis function were successfully estimated from the reflectance spectra of olivine [2]. The RBF network is widely applicable in a range of fields and can potentially be applied to constructing a general framework for science.

For hierarchical models such as the RBF network, Bayesian learning provides better generalization performance than maximum likelihood estimation. In Bayesian learning, two functions are considered to be the important indicators for estimation. One is the generalization error, which indicates the estimation accuracy for unknown data. The other is the stochastic complexity, which is used for model selection and optimization of hyperparameters. The coefficients  $\lambda$  of the leading terms for the asymptotic expansions of these functions are called the learning coefficient [3], and these values are model specific. Algebraic geometrical methods for hierarchical learning machines have also been established [4], and the values  $\lambda$  have been studied in various learning machines. However, the coefficients have not been

clarified in most models because the hierarchy of these models leads to difficulty in analysis.

We propose in this paper a novel method for calculating the learning coefficient by using the exchange Monte Carlo (EMC) method [5]. This proposal is based on the theoretical background in which the exchange ratio, which is calculated in the EMC simulation, depends on the learning coefficient and the setting of inverse temperatures [6]. Moreover, we calculate the learning coefficients in the RBF networks and verify the efficiency of our proposal by comparing the theoretical and experimental values.

This paper is organized into five sections. In Section 2, the RBF network and the general framework of Bayesian estimation are outlined. In Section 3, we propose our novel method for calculating the learning coefficient based on the EMC method. In Section 4, the results of the numerical analysis of the learning coefficient in the RBF networks are presented and discussed. Finally, a conclusion is given in Section 5.

## 2. Background

In this section, we introduce Bayesian learning in the radial basis function network.

### 2.1 Radial basis function (RBF) network

The RBF network is a regression model that is obtained by using the sum of basis functions as follows:

$$y = f(x; w) = \sum_{k=1}^K a_k \phi_k(x), \quad (1)$$

where  $\phi_k(x)$  is the radial basis function, which depends only on the distance from the center  $\mu_k$ . In this study, we take the following Gaussian functions  $\phi_k(x)$  as the basis functions,

<sup>1</sup> Graduate School of Frontier Science, The University of Tokyo, 5-1-5, Kashiwanoha, Kashiwa, Chiba, 277-8561, Japan

a) tokuda@mns.k.u-tokyo.ac.jp

b) nagata@mns.k.u-tokyo.ac.jp

c) okada@k.u-tokyo.ac.jp

$$\phi_k(x) = \exp\left(-\frac{b_k}{2}(x - \mu_k)^2\right). \quad (2)$$

The parameter set is  $w = \{a_k, \mu_k, b_k\}_{k=1}^K$ , where  $a_k$  and  $b_k$  are respectively the strength and the precision (the inverse of variance) of each basis function. The set of training samples  $D = \{X, Y\} = \{x_i, y_i\}_{i=1}^n$  consists of the individual pairs of input  $x_i$  and output  $y_i$ . The mean squared error function is defined by the training samples  $D$  and the fitting function  $f(x_i; w)$  as follows:

$$E(w) = \frac{1}{2n} \sum_{i=1}^n (y_i - f(x_i; w))^2. \quad (3)$$

This definition stands for the Gaussianity of the noise that is added to the output  $y_i$ ; it varies according to the process used to generate the data.

## 2.2 Bayesian estimation

The output  $y_i$  is assumed to be the sum of the true value  $f(x_i; w)$  and the noise  $\varepsilon_i$  as follows:

$$y_i = f(x_i; w) + \varepsilon_i, \quad (4)$$

where the noise  $\varepsilon_i$  is a random variable depending on the Gaussian distribution whose mean and variance are respectively 0 and  $\sigma^2$ . Given the input  $x_i$  and the parameter set  $w$ , the output  $y_i$  is given by the following conditional probability:

$$p(y_i | x_i, w) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i - f(x_i; w))^2}{2\sigma^2}\right). \quad (5)$$

For the independence of data, the probability density  $p(Y | X, w)$  of output set  $Y$  given input set  $X$  can be expressed as follows:

$$\begin{aligned} p(Y | X, w) &= \prod_{i=1}^n p(y_i | x_i, w) \\ &= \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left(-\frac{n}{\sigma^2} E(w)\right). \end{aligned} \quad (6)$$

In Bayesian estimation, the parameter  $w$  is regarded as a random variable, and the conditional probability density  $p(w | D)$  of parameter set  $w$  given training samples  $D$  is estimated based on the likelihood  $p(Y | X, w)$  and the density  $p(w)$ . Here  $p(w)$  and  $p(w | D)$  are respectively called the prior density and the posterior density. The posterior density  $p(w | D)$  can be expressed by using Bayes' theorem as follows:

$$\begin{aligned} p(w | D) &= \frac{p(Y | X, w)p(w)}{p(Y | X)} \\ &= \frac{1}{Z(D)} \exp\left(-\frac{n}{\sigma^2} E(w)\right) p(w), \end{aligned} \quad (7)$$

$$Z(D) = \int \exp\left(-\frac{n}{\sigma^2} E(w)\right) p(w) dw, \quad (8)$$

where  $Z(D)$  is a normalization constant called the marginal likelihood or the partition function. The function  $F(D)$  is called the stochastic complexity or the free energy and is defined as follows:

$$F(D) = -\log Z(D). \quad (9)$$

This function is used as the evaluation function for model selection and the optimization of the hyperparameters. Numerical integration by using the Markov chain Monte Carlo (MCMC) method

is a well-known way to calculate  $F(D)$  [7]. Bayesian learning enables us to estimate the true distribution  $q(y | x)$  from the predictive distribution  $p(y | x, D)$ , which is defined as the following function for unknown data  $(x, y)$ ,

$$p(y | x, D) = \int p(y | x, w) p(w | D) dw. \quad (10)$$

The gap between the true distribution  $q(y | x)$  and the predictive distribution  $p(y | x, D)$  is defined by using the Kullback distance as follows:

$$G(D) = \int q(y | x) q(x) \log \frac{q(y | x)}{p(y | x, D)} dx dy, \quad (11)$$

where  $G(D)$  is called the generalization error, and  $q(x)$  is the true distribution of input  $x$ .

## 2.3 Learning coefficient

The stochastic complexity  $F(D)$  and the generalization error  $G(D)$  can be expressed as the following asymptotic expansion for  $\frac{n}{\sigma^2} \rightarrow \infty$  [3],

$$F(D) \simeq \frac{n}{\sigma^2} E(\hat{w}) + \lambda \log \frac{n}{\sigma^2} + o\left(\frac{n}{\sigma^2}\right), \quad (12)$$

$$G(D) \simeq E(\hat{w}) + \frac{\sigma^2}{n} G_0(D) + o\left(\frac{\sigma^2}{n}\right), \quad (13)$$

where  $\hat{w}$  represents the maximum likelihood estimator of the parameter  $w$ , which minimizes the error function  $E(w)$ , and  $\lambda$  is a rational number called the learning coefficient, which is given as the absolute value of the largest pole of the following zeta function:

$$\zeta(z) = \int (E(w) - E(\hat{w}))^z p(w) dw. \quad (14)$$

The expectation value of the random variable  $G_0(D)$  over all sets of training samples is equal to  $\lambda$ . The learning coefficient  $\lambda$  represents how the true parameters exist in the parameter space and determines the speed that the predictive distribution  $p(y | x, D)$  converges towards the true distribution  $q(y | x)$ . Its value depends on the true distribution  $q(y | x)$ , the likelihood  $p(Y | X, w)$ , and the prior density  $p(w)$ . In the field of algebraic geometry,  $\lambda$  is called the real log canonical threshold (RLCT), which is well-known as an important value that represents the relative property of a pair of algebraic varieties [8]. In recent studies, algebraic geometrical analyses have been established for hierarchical learning machines. Using these analyses makes it possible to study the values  $\lambda$  in various learning machines, e.g. artificial neural networks [4] [9], Gaussian mixtures [10], reduced rank regressions [11], and Boltzmann machines [12]. However, there are many models whose values have not been clarified. The singularity of a model caused by hierarchy or exchange symmetry leads to difficulty in analysis. The coefficient  $\lambda = \frac{\dim(w)}{2}$  holds in models without singularity, and hence,  $F(D)$  is equal to the Bayesian Information Criterion (BIC), which is well-known as an approximate solution of  $F(D)$  [13], except for the constant  $o\left(\frac{n}{\sigma^2}\right)$  depending on the training samples.

## 3. Proposed method

In this section, we propose a novel method for calculating the learning coefficient by using the EMC method.

### 3.1 Exchange Monte Carlo (EMC) method

Local minima solutions often present a problem in the optimization of parameters. This problem can be resolved in principle by using the Markov chain Monte Carlo (MCMC) methods, which are a class of algorithms for sampling from target densities; these methods are based on constructing Markov chains. However, the calculation cost depends heavily on the target density or the initial state. The EMC method is a kind of MCMC method that provides a more effective solution for the problem of slow relaxation [5].

In the EMC simulation, parameter sampling is carried out from the following joint density  $p(w_1, \dots, w_L)$ :

$$p(w_1, \dots, w_L) = \prod_{l=1}^L p(w_l; \beta_l), \quad (15)$$

$$p(w_l; \beta_l) = \frac{1}{z(\beta_l)} \exp\left(-\frac{n\beta_l}{\sigma^2} E(w_l)\right) p(w_l), \quad (16)$$

$$z(\beta_l) = \int \exp\left(-\frac{n\beta_l}{\sigma^2} E(w_l)\right) p(w_l) dw_l, \quad (17)$$

where  $p(w_l; \beta_l)$  and  $z(\beta_l)$  respectively represent the posterior density and the marginal likelihood in each replica defined by different inverse temperatures  $\{\beta_l; l = 1, \dots, L\}$ . Sampling from the joint density  $p(w_1, \dots, w_L)$  is equivalent to sampling from each replica in parallel. In practice, we set inverse temperatures as  $0 = \beta_1 \leq \beta_2 \leq \dots \leq \beta_L = 1$ , so that  $p(w_1; \beta_1)$  and  $p(w_L; \beta_L)$  are respectively equal to the prior  $p(w)$  and the posterior  $p(w | D)$ .

The algorithm is constructed with the following update rules with which the joint density  $p(w_1, \dots, w_L)$  is invariant.

1. Update state in each replica  
Sample from each posterior density  $p(w_l; \beta_l)$  by using the Metropolis algorithm, one of the conventional MCMC methods, in parallel.
2. State exchange between two adjacent replicas  
Exchange the states  $w_l$  and  $w_{l+1}$  at every step according to the following probability  $u$ :

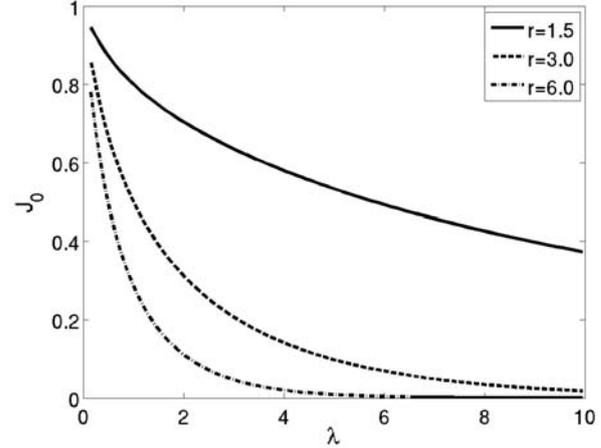
$$u(w_l, w_{l+1}; \beta_l, \beta_{l+1}) = \min(1, v(w_l, w_{l+1}; \beta_l, \beta_{l+1})), \quad (18)$$

$$\begin{aligned} v(w_l, w_{l+1}; \beta_l, \beta_{l+1}) &= \frac{p(w_{l+1}; \beta_l) p(w_l; \beta_{l+1})}{p(w_l; \beta_l) p(w_{l+1}; \beta_{l+1})} \\ &= \exp\left(\frac{n}{\sigma^2} (\beta_{l+1} - \beta_l) (E(w_{l+1}) - E(w_l))\right). \end{aligned} \quad (19)$$

### 3.2 Asymptotic behavior of average exchange ratio

The exchange ratio, which is calculated in the EMC simulation, represents sampling efficiency. Its value depends on the number  $n$  of data, the noise variance  $\sigma^2$ , the error function  $E(w)$ , and the setting of inverse temperatures  $\{\beta_l\}$ . The average exchange ratio  $J_0$  is defined as follows, and it converges in the low temperature limit, that is, as  $\frac{n}{\sigma^2} \rightarrow \infty$  [6]:

$$\begin{aligned} J_0 &= \int \int u(w_l, w_{l+1}; \beta_l, \beta_{l+1}) p(w_l; \beta_l) p(w_{l+1}; \beta_{l+1}) dw_l dw_{l+1} \\ &\rightarrow \frac{2r^\lambda \Gamma(2\lambda)}{\Gamma(\lambda)^2} \int_0^1 \frac{s^{\lambda-1}}{(r+s)^{2\lambda}} ds, \end{aligned} \quad (20)$$



**Fig. 1** Theoretical value of the average exchange ratio  $J_0$  for the value  $\lambda$ . The horizontal and vertical axes respectively represent the learning coefficient  $\lambda$  and the average exchange ratio  $J_0$ . The varied line styles indicate different values for temperature ratio  $r$ ; for the solid line,  $r = 1.5$ , for the dashed line,  $r = 3.0$ , and for the dash-dotted line,  $r = 6.0$ .

where  $r = \frac{\beta_{l+1}}{\beta_l}$ . By setting the inverse temperatures as a geometric progression, every temperature ratio  $r$  between each replica becomes equal. Therefore, it is guaranteed that every exchange ratio in the low temperature region will converge to the same constant. Figure 1 plots the theoretical value of the average exchange ratio  $J_0$  for the value  $\lambda$ . The horizontal axis and the vertical axis respectively represent the learning coefficient  $\lambda$  and the average exchange ratio  $J_0$ . The temperature ratio  $r$  varies with the line style; i.e., the solid line indicates an  $r$  value of 1.5, the dashed line indicates  $r$  of 3.0, and the dash-dotted line indicates  $r$  of 6.0. The average exchange ratio  $J_0$  is monotonically decreasing for the learning coefficient  $\lambda$  regardless of the temperature ratio  $r$ . This suggests that the value  $\lambda$  is determined from the exchange ratio  $J$  as the inverse function for sufficiently large  $\frac{n}{\sigma^2}$ .

### 3.3 Numerical analysis of learning coefficient

We propose a novel method for calculating the learning coefficient. This method is based on the theoretical background in which the exchange ratio, which is calculated in the EMC simulation, depends on the learning coefficient and the setting of inverse temperatures. The learning coefficient is calculated using the following procedure.

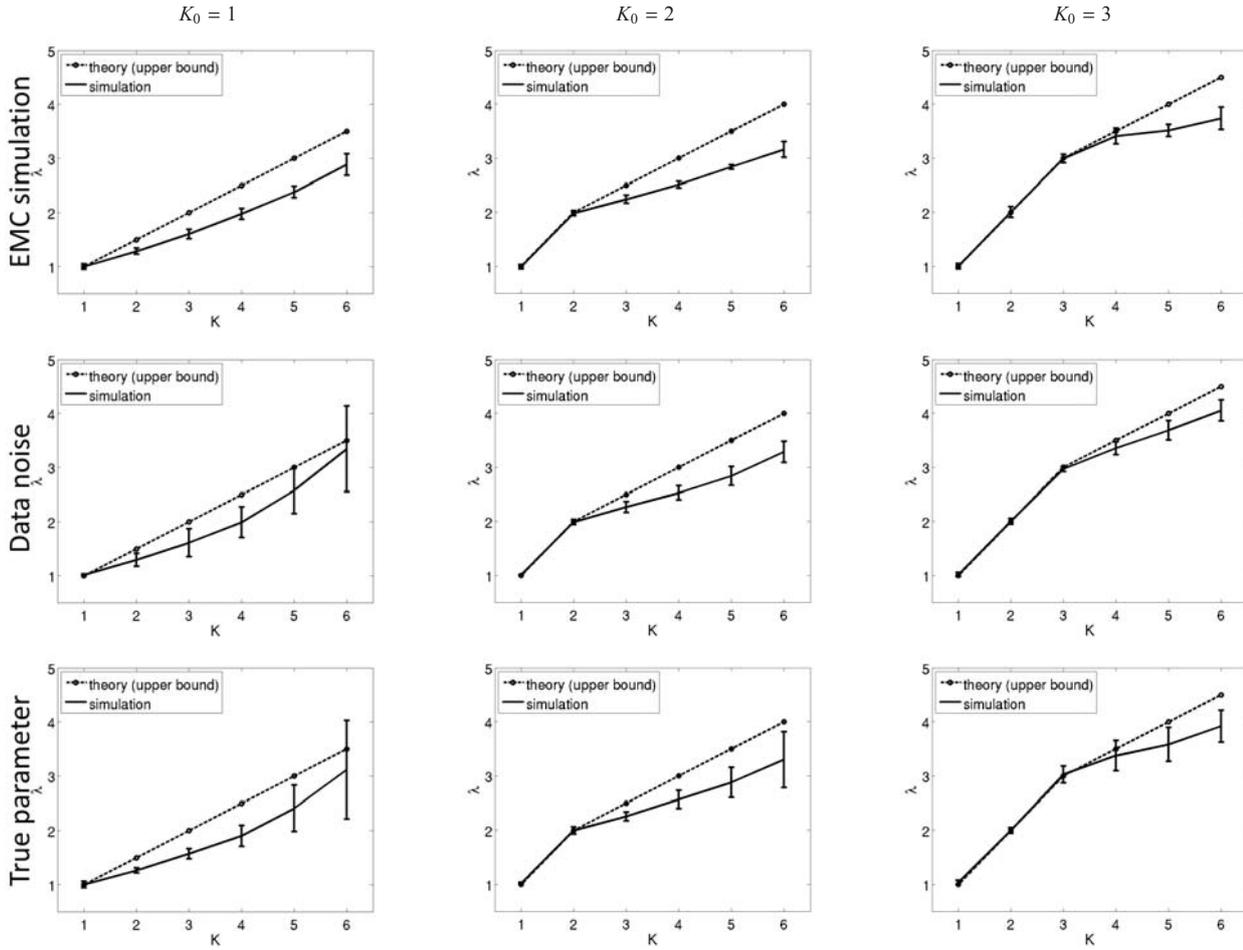
1. Setting of inverse temperatures  
Set the sequence of inverse temperatures as the following geometric progression:

$$\beta_l = \begin{cases} 0 & (\text{if } l = 1) \\ r^{l-L} & (\text{otherwise}), \end{cases} \quad (21)$$

where  $r$  is an arbitrary constant that satisfies  $r > 1$ . Note that  $0 \leq \beta_l \leq 1$  for all  $l = 1, \dots, L$ .

2. Calculation of the exchange ratio  
Simulate the learning by the EMC method and calculate the exchange ratio  $J$  as follows:

$$J = \frac{\alpha}{N}, \quad (22)$$



**Fig. 2** The theoretical upper bounds and the experimental values of the learning coefficient. The left, middle, and right graphs in each row respectively show the results of simulation for the training samples whose true number of bases was  $K_0 = 1$ ,  $K_0 = 2$ , and  $K_0 = 3$ . The upper, middle, and lower ones in each column respectively show the averages over five trials in which the following conditions were individually changed: the state exchange in the EMC method, the addition of noise to training samples, and the setting of the true parameter. In each graph, the horizontal and vertical axes respectively represent the number  $K$  of bases and the learning coefficient  $\lambda$ . The dashed line in each graph shows the theoretical upper bounds, and the solid line shows the average and the dispersion of the experimental values of the learning coefficient by using the proposed method. The error bars in these figures represent twice the standard deviation.

where  $N$  is the number of samples, and  $\alpha$  is the acceptance frequency between the replicas defined by the inverse temperatures  $\beta_L$  and  $\beta_{L-1}$ .

3. Numerical resolution of the learning coefficient  
Substitute the temperature ratio  $r$  and the exchange ratio  $J$  for Eq. (20), and calculate back to the learning coefficient  $\lambda$  by using the bisection method.

## 4. Simulation and discussion

In this section, we calculate the learning coefficients in the RBF networks and discuss the accuracy and effectiveness of the proposed method by comparing the experimental values and the theoretical upper bounds.

### 4.1 Settings

Input  $x$  was taken from the range [150 – 175] in steps of 0.1 for every training sample, and the total number  $n$  was 251. The noise variance was set as  $\sigma^2 = 1$ . The training samples were generated

from the following true function:

$$g(x; w^*) = \sum_{k=1}^{K_0} a_k^* \exp\left(-\frac{b_k^*}{2}(x - \mu_k^*)^2\right), \quad (23)$$

where  $K_0$  represents the true number of bases, and  $w^* = \{a_k^*, \mu_k^*, b_k^*\}_{k=1}^{K_0}$  represents the true parameters.

We defined the prior density  $p(w)$  as follows:

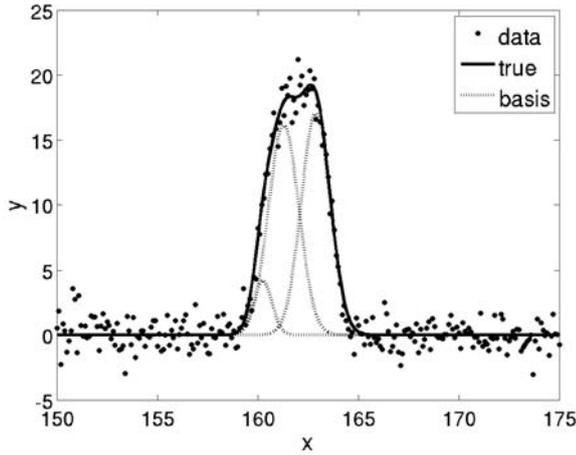
$$p(w) = \prod_{k=1}^K \varphi(a_k) \varphi(\mu_k) \varphi(b_k), \quad (24)$$

$$\begin{aligned} \varphi(a_k) &= \text{Gamma}(a_k; \eta_a, \theta_a) \\ &= \frac{1}{\Gamma(\eta_a)} \theta_a^{\eta_a} a_k^{\eta_a-1} \exp(-\theta_a a_k), \end{aligned} \quad (25)$$

$$\begin{aligned} \varphi(\mu_k) &= \mathcal{N}(\mu_k; \nu_0, \xi_0^{-1}) \\ &= \sqrt{\frac{\xi_0}{2\pi}} \exp\left(-\frac{\xi_0}{2}(\mu_k - \nu_0)^2\right), \end{aligned} \quad (26)$$

$$\varphi(b_k) = \text{Gamma}(b_k; \eta_b, \theta_b), \quad (27)$$

where the hyperparameters were  $\eta_a = 5$ ,  $\theta_a = 0.25$ ,  $\nu_0 = 162.5$ ,



**Fig. 3** The training samples generated for model selection. The horizontal and vertical axes respectively represent the input  $x$  and the output  $y$ . The true number of bases is  $K_0 = 3$ . The black dots represent the data. The solid and dotted lines respectively indicate the true function  $g(x)$  and each basis function.

$\xi_0 = 2$ ,  $\eta_b = 4$ , and  $\theta_b = 1$ . The candidate model size for the estimation was set as the range from  $K = 1$  to  $K = 6$ .

The number of inverse temperatures was  $L = 32$ , and their ratios were all  $r = 1.5$ . The initial state of each parameter  $w_j$  and the state update of  $w_1$  for every step were determined according to the density  $p(w)$ . The iteration was set as 50,000 steps for the burn-in period, and 50,000 steps for the calculation of the exchange ratio.

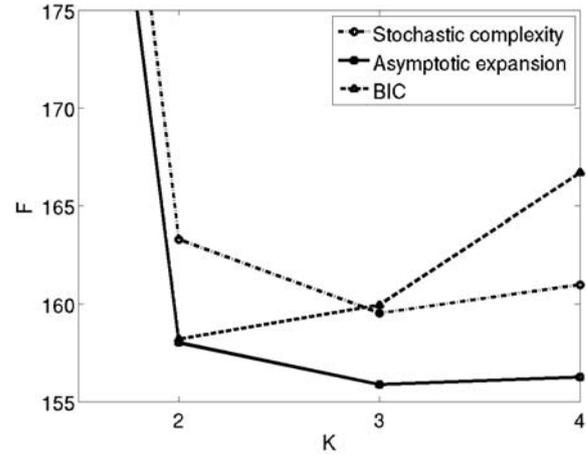
#### 4.2 Accuracy evaluation of learning coefficient

We calculated the learning coefficients  $\lambda$  in variance-fixed RBF networks by using the proposed method to compare their values with their clarified theoretical upper bounds. Here the variance-fixed RBF network was an RBF network whose variance of each basis function all had an equal value,  $b_k = b_k^* = 5.67$ . The theoretical upper bound of the learning coefficient in the variance-fixed RBF network is considered to be the same as the analytical solution for Gaussian mixtures as follows [10]:

$$\lambda \leq \begin{cases} \frac{\dim(w)}{2} = K & (K \leq K_0) \\ \frac{\dim(w^*)}{2} + \frac{K - K_0}{2} = \frac{K + K_0}{2} & (K > K_0). \end{cases} \quad (28)$$

In practice, the values  $\lambda$  are assumed to have dispersion caused by the state exchange in the EMC method, the noise added to the training samples, and the setting of the true parameters. On the basis of this viewpoint, we individually changed the above conditions and evaluated the accuracy of the learning coefficients. The training samples were generated from the true parameters  $w^*$  according to the prior density  $p(w)$ .

The experimental values and the theoretical upper bounds of the learning coefficient are shown in Fig. 2. The left, middle, and right figures in each row respectively show the simulation results for the training samples whose true number of bases was  $K_0 = 1$ ,  $K_0 = 2$ , and  $K_0 = 3$ . The upper, middle, and lower ones in each column respectively show the averages over five trials in which the following conditions were individually changed: the state exchange in the EMC method, the addition of noise to train-



**Fig. 4** Results of model selection for the data shown in Fig. 3. The horizontal and vertical axes respectively represent the number  $K$  of bases and the values of the evaluation functions. The dash-dotted line, solid line, and dashed line respectively show the stochastic complexity, its asymptotic expansion (Eq. (12)), and BIC. The stochastic complexity and its asymptotic expansion take the minimum value at  $K = 3$ , while BIC does so at  $K = 2$ .

ing samples, and the setting of the true parameters. In each graph, the horizontal axis and the vertical axis respectively represent the number  $K$  of bases and the learning coefficient  $\lambda$ . The dashed line shows the theoretical upper bounds, and the solid line shows the average and the dispersion of the experimental values of the learning coefficient using the proposed method. The error bars in these graphs represent twice the standard deviation.

Every average value is below the theoretical upper bound, and their dispersions are sufficiently small. This suggests that the proposed method provides an accurate learning coefficient. For  $K \leq K_0$ , the experimental values approach the theoretical values, and their dispersions are significantly small. This shows that the singularity of the model rarely affects such  $K$  values. For  $K > K_0$ , on the other hand, the dispersion tends to be larger depending on the increase in  $K$ . This shows that the larger dimension of the parameter space constructs a more complex singular structure.

#### 4.3 Application to model selection

We applied the proposed method to model selection by using the asymptotic expansion of the stochastic complexity and then evaluated how effective our method was. The training samples were generated from the following true parameters:

$$\begin{pmatrix} a_1^* \\ a_2^* \\ a_3^* \end{pmatrix} = \begin{pmatrix} 5.6920 \\ 17.0239 \\ 16.1475 \end{pmatrix}, \begin{pmatrix} \mu_1^* \\ \mu_2^* \\ \mu_3^* \end{pmatrix} = \begin{pmatrix} 160.2421 \\ 162.8885 \\ 161.2859 \end{pmatrix}, \\ \begin{pmatrix} b_1^* \\ b_2^* \\ b_3^* \end{pmatrix} = \begin{pmatrix} 4.7914 \\ 1.9104 \\ 1.7696 \end{pmatrix}.$$

Figure 3 plots the generated training samples, and Fig. 4 shows the results of model selection by using the stochastic complexity, its asymptotic expansion (Eq. (12)), and BIC. The horizontal and vertical axes respectively represent the number  $K$  of bases and the values of the evaluation functions. The dash-dotted line, solid line, and dashed line respectively show the stochastic complex-

ity, its asymptotic expansion (Eq. (12)), and BIC. The stochastic complexity  $F(D)$  and the minimum error  $E(\hat{w})$  were calculated in the EMC simulation.

BIC takes the minimum value at  $K = 2$ ; in other words, the wrong model size was selected. This is because the learning coefficient in the singular model is  $\lambda < \frac{\dim w}{2}$ , so the penalty term is estimated as being higher than the true value. This result shows the risk that BIC may estimate a smaller number  $K$  of bases than the true number  $K_0$ . On the contrary, the stochastic complexity and its asymptotic expansion Eq. (12) take the minimum value at  $K = 3$ ; that is, the true model size was selected. Furthermore, their values are considered to be equal, except for the constant  $o\left(\frac{n}{\sigma^2}\right)$ , depending on the training samples. This reveals that both criteria provide the same model selection result thanks to the exact estimation of the learning coefficient  $\lambda$  and the minimum error  $E(\hat{w})$ .

## 5. Conclusion

We proposed a novel method for calculating a learning coefficient by using the exchange Monte Carlo method and discussed the method's accuracy in a simulation for an RBF network. The accuracy of the learning coefficient calculated by using our proposed method in a variance-fixed RBF network was shown to be valid by comparing the results with the theoretical upper bound. Moreover, we applied the proposed method to model selection by using the asymptotic expansion of the stochastic complexity and verified that our method was effective.

**Acknowledgments** This work was supported by the Ministry of Education, Science, Sports and Culture in Japan, Grant-in-Aid for Scientific Research 20240020, 22700230 and 24654118.

## References

- [1] Rei Muraoka, Kenji Nagata, Takehiko Sasaki and Masato Okada, Application of Bayesian Estimation for XPS Data Analysis, IEICE Technical Report, 110(476), 125-130, 2011.
- [2] Kenji Nagata, Seiji Sugita, and Masato Okada, Bayesian spectral deconvolution with the exchange Monte Carlo method, Neural Networks, 28, 82-89, 2012.
- [3] Sumio Watanabe, Algebraic Analysis for Nonidentifiable Learning Machines, Neural Computation, 13(4), 899-933, 2001.
- [4] Sumio Watanabe, Algebraic geometrical methods for hierarchical learning machines, Neural Networks, 14(8), 1049-1060, 2001.
- [5] Koji Hukushima and Koji Nemoto, Exchange Monte Carlo Method and Application to Spin Glass Simulations, Journal of the Physical Society of Japan, 65(6), 1604-1608, 1996.
- [6] Kenji Nagata and Sumio Watanabe, Asymptotic Behavior of Exchange Ratio in Exchange Monte Carlo Method, Neural Networks, 21(7), 980-988, 2008.
- [7] Yoshihiko Ogata, A Monte Carlo method for an objective Bayesian procedure, Annals of the Institute of Statistical Mathematics, 42(3), 403-433, 1990.
- [8] I. N. Bernshtein, The analytic continuation of generalized functions with respect to a parameter, Functional Analysis and Its Applications, 6(4), 273-285, 1972.
- [9] Miki Aoyagi and Kenji Nagata, Learning coefficient of generalization error in Bayesian estimation and vandermonde matrix-type singularity, Neural Computation, 24(6), 1569-1610, 2012.
- [10] Keisuke Yamazaki and Sumio Watanabe, Singularities in mixture models and upper bounds of stochastic complexity, International Journal of Neural Networks, 16, 1029-1038, 2003.
- [11] Miki Aoyagi and Sumio Watanabe, Stochastic complexities of reduced rank regression in Bayesian estimation, Neural Networks, 18(7), 924-933, 2005.
- [12] Keisuke Yamazaki and Sumio Watanabe, Singularities in complete bipartite graph-type Boltzmann machines and upper bounds of stochastic complexities, IEEE Transactions on Neural Networks, 16(2), 312-324, 2005.
- [13] Gideon Schwarz, Estimating the dimension of a model, The Annals of Statistics, 6, 461-464, 1978.