IPSJ Transactions on Mathematical Modeling and Its Applications Vol.6 No.3 117–123 (Dec. 2013)

Regular Paper

A Numerical Analysis of Learning Coefficient in Radial Basis Function Network

Satoru Tokuda[1] Kenji Nagata[1] Masato Okada[1,a)]

Received: January 30, 2013, Revised: March 20, 2013,
Accepted: June 7, 2013

Abstract: The radial basis function (RBF) network is a regression model that uses 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 λ 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 λ 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:

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

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

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

[1] Graduate School of Frontier Science, The University of Tokyo, Kashiwa, Chiba 277–8561, Japan
a) okada@k.u-tokyo.ac.jp

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:

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

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 purpose of the Bayesian estimation is to evaluate the parameter set not as the determined variable but as the probability distribution. For hierarchical learning machines such as the RBF network, Bayesian estimation provides better generalization performance than point estimation such as maximum likelihood estimation and maximum a posteriori estimation.

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

$$y_i = f(x_i; w) + \varepsilon_i, \tag{4}$$

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

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

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

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

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

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

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

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:

$$F(D) = -\log Z(D). \tag{9}$$

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 $q(y \mid x)$ from the predictive distribution $p(y \mid x, D)$, which is defined as the following function for unknown data (x, y),

$$p(y \mid x, D) = \int p(y \mid x, w)p(w \mid D)\mathrm{d}w. \tag{10}$$

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

$$G(D) = \int q(y \mid x)q(x)\log\frac{q(y \mid x)}{p(y \mid x, D)}\mathrm{d}x\mathrm{d}y, \tag{11}$$

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

2.3 Learning Coefficient

The average log loss function $H(w)$ is defined as follows:

$$H(w) = -\int q(y \mid x)q(x)\log p(y \mid x, w)\mathrm{d}x\mathrm{d}y. \tag{12}$$

Note that the parameter w_0 which minimizes $H(w)$ is not equal to the maximum likelihood estimator for hierarchical learning machines. The stochastic complexity $F(D)$ and the generalization error $G(D)$ can be expressed as the following asymptotic expansion for $\frac{n}{\sigma^2} \to \infty$ [3],

$$F(D) = \frac{n}{\sigma^2}E(w_0) + \lambda\log\frac{n}{\sigma^2} + O_p\left(\log\log\frac{n}{\sigma^2}\right), \tag{13}$$

$$G(D) = E(w_0) + \lambda\left(\frac{n}{\sigma^2}\right)^{-1} + o_p\left(\left(\frac{n}{\sigma^2}\right)^{-1}\right), \tag{14}$$

where λ is a rational number called the learning coefficient, which is given as the absolute value of the largest pole of the following zeta function:

$$\zeta(z) = \int (H(w) - H(w_0))^z p(w)\mathrm{d}w. \tag{15}$$

The learning coefficient λ represents how the true parameters exist in the parameter space and determines the speed that the predictive distribution $p(y \mid x, D)$ converges towards the true distribution $q(y \mid x)$. Its value depends on the true distribution $q(y \mid x)$, the likelihood $p(Y \mid X, w)$, and the prior density $p(w)$. In the field of algebraic geometry, λ 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 λ 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 leads to difficulty in analysis. In general, a learning machine

is called regular if a parameter set is uniquely determined for a distribution and if its Fisher information matrix is always positive definite. If otherwise, it is called singular. The RBF networks with K bases are regular for the case that $K \leq K_0$, and are singular for the case that $K > K_0$. The relationship $\lambda = \frac{\dim(w)}{2}$ holds with respect to the coefficient λ in regular models, 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 random variable $O_p(1)$ 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 $p(w_1, \ldots, w_L)$:

$$p(w_1, \ldots, w_L) = \prod_{l=1}^{L} p(w_l; \beta_l), \tag{16}$$

$$p(w_l; \beta_l) = \frac{1}{z(\beta_l)} \exp\left(-\frac{n\beta_l}{\sigma^2} E(w_l)\right) p(w_l), \tag{17}$$

$$z(\beta_l) = \int \exp\left(-\frac{n\beta_l}{\sigma^2} E(w_l)\right) p(w_l) \mathrm{d}w, \tag{18}$$

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 $\{\beta_l; l = 1, \ldots, L\}$. Sampling from the joint density $p(w_1, \ldots, w_L)$ is equivalent to sampling from each replica in parallel. In practice, we set inverse temperatures as $0 = \beta_1 \leq \beta_2 \leq \cdots \leq \beta_L = 1$, so that $p(w_1; \beta_1)$ and $p(w_L; \beta_L)$ are respectively equal to the prior $p(w)$ and the posterior $p(w \mid D)$.

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

$$u(w_l, w_{l+1}; \beta_l, \beta_{l+1}) = \min(1, v(w_l, w_{l+1}; \beta_l, \beta_{l+1})), \tag{19}$$

$$v(w_l, w_{l+1}; \beta_l, \beta_{l+1}) = \frac{p(w_{l+1}; \beta_l) p(w_l; \beta_{l+1})}{p(w_l; \beta_l) p(w_{l+1}; \beta_{l+1})} = \exp\left(\frac{n}{\sigma^2}(\beta_{l+1} - \beta_l)(E(w_{l+1}) - E(w_l))\right). \tag{20}$$

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 of data n, the noise variance σ^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 $\frac{n}{\sigma^2} \to \infty$ [6]:

$$J_0 = \iint u(w_l, w_{l+1}; \beta_l, \beta_{l+1}) p(w_l; \beta_l) p(w_{l+1}; \beta_{l+1}) \mathrm{d}w_l \mathrm{d}w_{l+1} \to \frac{2r^\lambda \Gamma(2\lambda)}{\Gamma(\lambda)^2} \int_0^1 \frac{s^{\lambda-1}}{(r+s)^{2\lambda}} \mathrm{d}s, \tag{21}$$

where $r = \frac{\beta_{l+1}}{\beta_l}$. If the temperature ratio r is fixed, J_0 is the single-valued function as a function of λ. 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 λ. The horizontal axis and the vertical axis respectively represent the learning coefficient λ 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 λ regardless of the temperature ratio r. This suggests that the value λ 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.

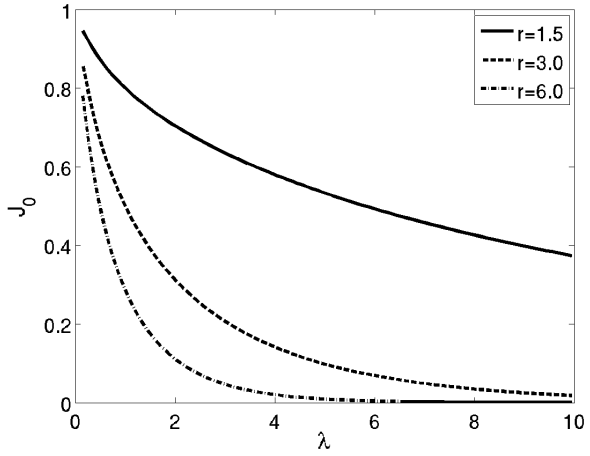

Fig. 1 Theoretical value of the average exchange ratio J_0 for the value λ. The horizontal and vertical axes respectively represent the learning coefficient λ and the average exchange ratio J_0. The varied line styles indicate different values for temperature ratio r; for the solid line, $r = 1.5$, for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

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

$$\beta_l = \begin{cases} 0 & (\text{if} \quad l = 1) \\ r^{l-L} & (\text{otherwise}), \end{cases} \tag{22}$$

   where r is an arbitrary constant that satisfies $r > 1$. Note that $0 \le \beta_l \le 1$ for all $l = 1, \ldots, L$.
2. Calculation of the exchange ratio
   Simulate the learning by the EMC method and calculate the exchange ratio J as follows:

$$J = \frac{\alpha}{N}, \tag{23}$$

   where N is the iteration for the EMC algorithm except the burn-in period, and α is the acceptance frequency between the replicas defined by the inverse temperatures β_L and β_{L-1}.
3. Numerical resolution of the learning coefficient
   Substitute the temperature ratio r and the exchange ratio J for Eq. (21), and calculate back to the learning coefficient λ 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:

$$g(x; w^*) = \sum_{k=1}^{K_0} a_k^* \exp\left(-\frac{b_k^*}{2}(x - \mu_k^*)^2\right), \tag{24}$$

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:

$$p(w) = \prod_{k=1}^{K} \varphi(a_k)\varphi(\mu_k)\varphi(b_k), \tag{25}$$

$$\varphi(a_k) = \text{Gamma}(a_k; \eta_a, \theta_a) = \frac{1}{\Gamma(\eta_a)} \theta_a{}^{\eta_a} a_k{}^{\eta_a - 1} \exp(-\theta_a a_k), \tag{26}$$

$$\varphi(\mu_k) = \mathcal{N}(\mu_k; \nu_0, \xi_0{}^{-1}) = \sqrt{\frac{\xi_0}{2\pi}} \exp\left(-\frac{\xi_0}{2}(\mu_k - \nu_0)^2\right), \tag{27}$$

$$\varphi(b_k) = \text{Gamma}(b_k; \eta_b, \theta_b), \tag{28}$$

where the hyperparameters were $\eta_a = 5$, $\theta_a = 0.25$, $\nu_0 = 162.5$, $\xi_0 = 2$, $\eta_b = 4$, 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_l 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 Calculation and Accuracy Evaluation

We calculated the learning coefficients λ in variance-fixed RBF networks by using the proposed method to compare their clarified theoretical and experimental values. Here the variance-fixed RBF network was an RBF network whose variance of each basis function all had an equal value, $b_k = b_k^* = 5.67$. For $K \le K_0$, this model is regular, so that the exact solution of the learning coefficient is given as follows:

$$\lambda = \frac{\dim(w)}{2} = K \qquad (K \le K_0). \tag{29}$$

For $K > K_0$, on the other hand, the model is singular, and the upper bound of the learning coefficient is considered to be the same as the analytical solution for Gaussian mixtures as follows [10]:

$$\lambda \le \frac{\dim(w^*)}{2} + \frac{K - K_0}{2} = \frac{K + K_0}{2} \qquad (K > K_0). \tag{30}$$

In practice, the values λ 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 theoretical and experimental values 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 graphs in each column show the average and the dispersion over the EMC simulation 5 times. The middle and lower ones respectively show the average over the data noise and over the true parameter. In each graph, the horizontal axis and the vertical axis respectively represent the number of bases K and the learning coefficient λ. The dashed line shows the theoretical values, which indicate the exact solutions for $K \le K_0$ and the upper bounds for $K > K_0$. 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.

For $K \le K_0$, the experimental values approach the exact solutions, and their dispersions are significantly small. This shows that the proposed method provides an accurate learning coefficient. For $K > K_0$, on the other hand, every average value is below the upper bound, and 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. We cannot analytically derive the coefficient λ from its definition without knowing the true distribution and its model size. In contrast, the proposed method does not require such information, on the contrary, we can inversely estimate the model size from the behavior of the expectation value of the values λ over all sets of training samples.

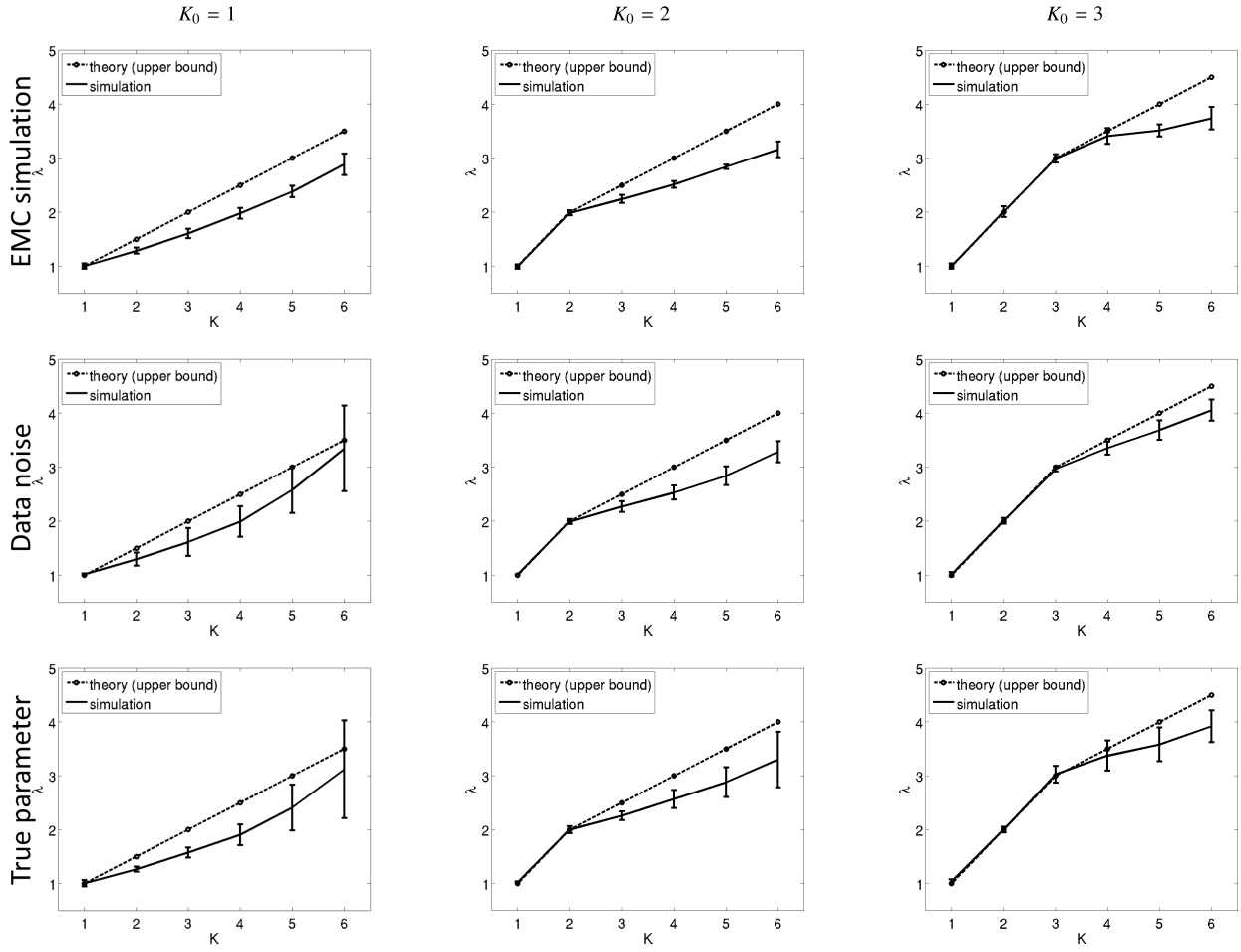

Fig. 2 The theoretical and experimental values of the learning coefficient. The left, middle, and right graphs in each row respectively show the results of simulation for the training samples whose true number of bases was $K_0 = 1$, $K_0 = 2$, and $K_0 = 3$. The upper graphs in each column show the average and the dispersion over the EMC simulation 5 times. The middle and lower ones respectively show the average over the data noise and over the true parameter. In each graph, the horizontal and vertical axes respectively represent the number of bases K and the learning coefficient λ. The dashed line in each graph shows the theoretical values, which indicate the exact solutions for $K \leq K_0$ and the upper bounds for $K > K_0$. 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.

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:

$$\begin{pmatrix} a_1^* \\ a_2^* \\ a_3^* \end{pmatrix} = \begin{pmatrix} 5.6920 \\ 17.0239 \\ 16.1475 \end{pmatrix}, \begin{pmatrix} \mu_1^* \\ \mu_2^* \\ \mu_3^* \end{pmatrix} = \begin{pmatrix} 160.2421 \\ 162.8885 \\ 161.2859 \end{pmatrix},$$

$$\begin{pmatrix} b_1^* \\ b_2^* \\ b_3^* \end{pmatrix} = \begin{pmatrix} 4.7914 \\ 1.9104 \\ 1.7696 \end{pmatrix}.$$

The left side of **Fig. 3** plots the generated training samples, and the right side shows the results of model selection by using the stochastic complexity, its asymptotic expansion (Eq. (13)), and BIC. The horizontal and vertical axes respectively represent the number of bases K 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. (13)), and BIC. In practice, the value $E(w_0)$ was calculated as the minimum error; we cannot know the true distribution. The stochastic complexity $F(D)$ 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 than the true number K_0. On the contrary, the stochastic complexity and its asymptotic expansion Eq. (13) take the minimum value at $K = 3$; that is, the true model size was selected. Furthermore, the values of the stochastic complexity and its asymptotic expansion are considered to be equal, except for the random variable $O_p\left(\log\log\frac{n}{\sigma^2}\right)$ 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 λ.

In addition, we checked the estimated parameter to verify the validity of above discussion. The result of maximum likelihood estimation is shown in **Fig. 4**. The left and right graphs respec-

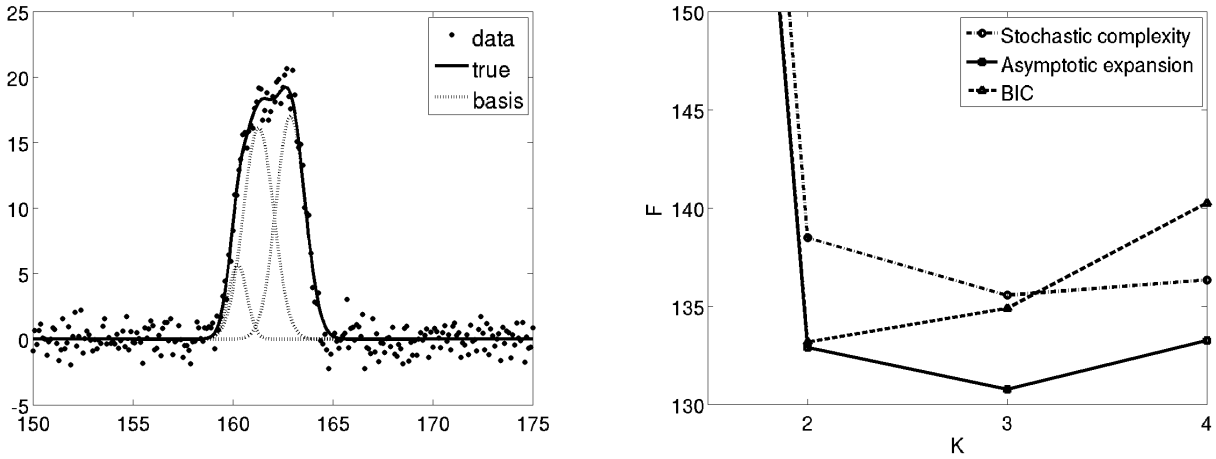

Fig. 3 Results of model selection. The left and right graphs respectively show the training samples and the values of criteria. (Left) The horizontal and vertical axes respectively represent the input x and the output y. The true number of bases is $K_0 = 3$. The black dots represent the data. The solid and dotted lines respectively indicate the true function $g(x)$ and each basis function. (Right) The horizontal and vertical axes respectively represent the number of bases K 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. (13)), and BIC. The stochastic complexity and its asymptotic expansion take the minimum value at $K = 3$, while BIC does so at $K = 2$.

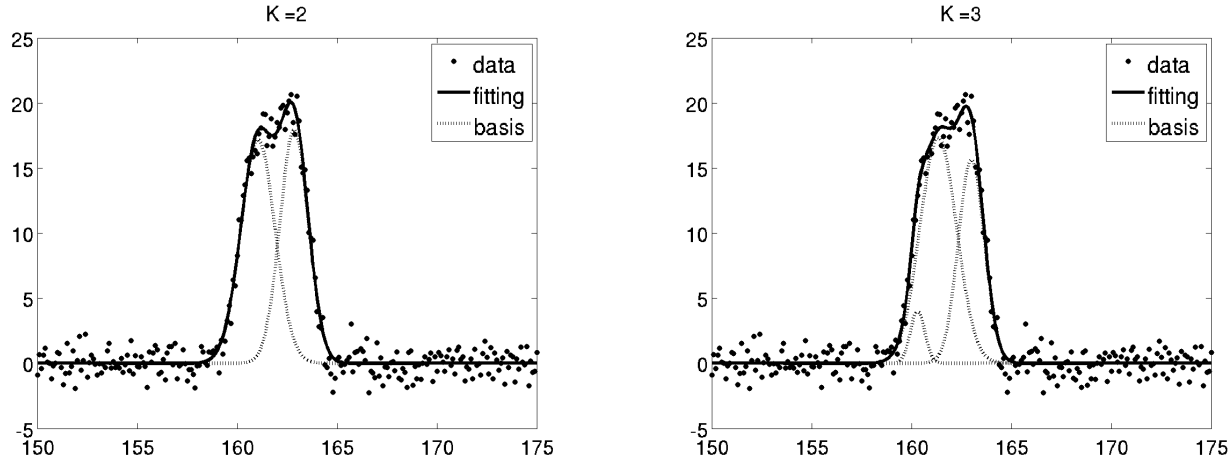

Fig. 4 Results of maximum likelihood estimation. The left and right graphs respectively show the fitting curves and the bases by using the RBF networks whose number of basis are $K = 2$ and $K = 3$.

tively show the fitting curves and the bases by using the RBF networks whose number of basis are $K = 2$ and $K = 3$. The result for $K = 3$ expresses the character of each basis of the true function better than the result for $K = 2$.

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 values. 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, Culture, Sports, Science and Technology in Japan, Grant-in-Aid for Scientific Research 22700230, 24654118, 25106506, 25120009, and 25330283.

References

[1] Muraoka, R., Nagata, K., Sasaki, T. and Okada, M.: Application of Bayesian Estimation for XPS Data Analysis, *IEICE Technical Report*, Vol.110, No.476, pp.125–130 (2011).
[2] Nagata, K., Sugita, S. and Okada, M.: Bayesian spectral deconvolution with the exchange Monte Carlo method, *Neural Networks*, Vol.28, pp.82–89 (2012).
[3] Watanabe, S.: *Algebraic Geometry and Statistical Learning Theory*, Cambridge University Press, Cambridge (2009).
[4] Watanabe, S.: Algebraic geometrical methods for hierarchical learning machines, *Neural Networks*, Vol.14, No.8, pp.1049–1060 (2001).
[5] Hukushima, K. and Nemoto, K.: Exchange Monte Carlo Method and Application to Spin Glass Simulations, *Journal of the Physical Society of Japan*, Vol.65, No.6, pp.1604–1608 (1996).
[6] Nagata, K. and Watanabe, S.: Asymptotic Behavior of Exchange Ratio in Exchange Monte Carlo Method, *Neural Networks*, Vol.21, No.7, pp.980–988 (2008).
[7] Ogata, Y.: A Monte Carlo method for an objective Bayesian procedure, *Annals of the Institute of Statistical Mathematics*, Vol.42, No.3, pp.403–433 (1990).
[8] Bernshtein, I.N.: The analytic continuation of generalized functions with respect to a parameter, *Functional Analysis and Its Applications*,

Vol.6, No.4, pp.273–285 (1972).
[9] Aoyagi, M. and Nagata, K.: Learning coefficient of generalization error in Bayesian estimation and vandermonde matrix-type singularity, *Neural Computation*, Vol.24, No.6, pp.1569–1610 (2012).
[10] Yamazaki, K. and Watanabe, S.: Singularities in mixture models and upper bounds of stochastic complexity, *International Journal of Neural Networks*, Vol.16, pp.1029–1038 (2003).
[11] Aoyagi, M. and Watanabe, S.: Stochastic complexities of reduced rank regression in Bayesian estimation, *Neural Networks*, Vol.18, No.7, pp.924–933 (2005).
[12] Yamazaki, K. and Watanabe, S.: Singularities in complete bipartite graph-type Boltzmann machines and upper bounds of stochastic complexities, *IEEE Trans. Neural Networks*, Vol.16, No.2, pp.312–324 (2005).
[13] Schwarz, G.: Estimating the dimension of a model, *The Annals of Statistics*, Vol.6, pp.461–464 (1978).

Satoru Tokuda received his B.Sc. degree from Yokohama City University, Yokohama, Japan, in 2012. He is currently working towords an M.Sc. degree in complexity science and engineering at The University of Tokyo, Kashiwa, Japan. His research interests mainly concern methodology of inverse problems, in particular, spectroscopic data analysis.

Kenji Nagata received his M.E. and Ph.D. degrees in computational intelligence and systems science from Tokyo Institute of Technology, Japan, in 2006 and 2008 respectively. He is currently an Assistant Professor at the Graduate School of Frontier Science, The University of Tokyo. His research interests include learning theory and theory and application of Markov chain Monte Carlo method.

Masato Okada received his M.Sc. and Ph.D. degrees from Osaka University, Osaka, Japan, in 1987 and 1997, respectively. From 1987 to 1989, he worked at Mitsubishi Electric Corporation. From 1991 to 1996 he was a Research Associate at Osaka University. He was a Researcher in the Kawato Dynamic Brain Project until 2001. He was a Deputy Laboratory Head in RIKEN Brain Science Institute, Saitama, Japan, until 2004. He is currently a Professor at the Graduate School of Frontier Science, The University of Tokyo. His research interests include computational aspects of neural networks and statistical mechanics for information processing.