PAPER

# A Sparse Modeling Method Based on Reduction of Cost Function in Regularized Forward Selection

**Katsuyuki HAGIWARA**[†a)], ***Member***

**SUMMARY** Regularized forward selection is viewed as a method for obtaining a sparse representation in a nonparametric regression problem. In regularized forward selection, regression output is represented by a weighted sum of several significant basis functions that are selected from among a large number of candidates by using a greedy training procedure in terms of a regularized cost function and applying an appropriate model selection method. In this paper, we propose a model selection method in regularized forward selection. For the purpose, we focus on the reduction of a cost function, which is brought by appending a new basis function in a greedy training procedure. We first clarify a bias and variance decomposition of the cost reduction and then derive a probabilistic upper bound for the variance of the cost reduction under some conditions. The derived upper bound reflects an essential feature of the greedy training procedure; i.e., it selects a basis function which maximally reduces the cost function. We then propose a thresholding method for determining significant basis functions by applying the derived upper bound as a threshold level and effectively combining it with the leave-one-out cross validation method. Several numerical experiments show that generalization performance of the proposed method is comparable to that of the other methods while the number of basis functions selected by the proposed method is greatly smaller than by the other methods. We can therefore say that the proposed method is able to yield a sparse representation while keeping a relatively good generalization performance. Moreover, our method has an advantage that it is free from a selection of a regularization parameter.

***key words:*** *regularized forward selection, nonparametric regression, sparse representation, thresholding method, cross validation*

## 1. Introduction

This paper considers a regression method using a linear combination of a large number of basis functions. This can be viewed as a nonparametric regression problem. In this setting, we need to suppress model complexity in order to achieve good generalization performance. There are two approaches for this purpose: smoothing by regularization and keeping a sparse representation. Of course, both are combined in some methods. In this paper, we focus on methods for obtaining a sparse representation. There are three different types of such methods: introduction of a specific cost function with parameters, backward elimination and forward selection. The support vector machine (SVM) obtains a sparse representation through an optimization procedure under a cost function that consists of a hinge loss function and an $\ell_2$-type regularizer (see, e.g., [7]). For this purpose, LASSO [23] employs a cost function that consists of squared error loss and an $\ell_1$-type regularizer. Unlike SVM, least squares SVM (LS-SVM) [22] or, equivalently, regularized least squares (RLS) [20] employ squared error as a loss and an $\ell_2$-type regularizer as a penalty in the cost function. The relevance vector machine (RVM) [24] is similar to those but is formulated in a Bayesian framework. In these methods, the regularizer controls the smoothness of output and sparseness is determined by a pruning method under an appropriate heuristic criterion (see, e.g., [22], [24]). These are backward elimination methods. Backward elimination generally features a large computational cost since a large model should be handled in training with the elimination procedure. On the other hand, forward selection employs a greedy training procedure and a model selection strategy. This procedure necessarily obtains a sparse representation without handling a large model; i.e. in the greedy training procedure, it starts from a simple model and adds a new basis function at each step.

Regularized forward selection (RFS) [18] is a forward selection method that employs a greedy training procedure under a regularized cost function. The greedy procedure in RFS is formulated to be suitable for an iterative calculation. The cost function in RFS consists of the squared error loss and an $\ell_2$ regularizer. This includes the least squares type of greedy procedure as a special case. The greedy procedure considered here selects a new basis function at each step in terms of the reduction of the regularized cost function. Orthogonal least squares (OLS) [5], [6] is a special version of RFS, in which candidates of an appended basis function are orthogonalized at each greedy step. This orthogonalization process makes the iterative calculation easy. As a model selection method for OLS, [5], [6] applied the prediction sum of squares (PRESS) statistics [3], [21], which is also called the leave-one-out cross validation (LOOCV) error. On the other hand, for training multi-layer perceptron, some extensions of the extreme learning machine (ELM) [28] also employ greedy procedures [9], [16], [17]. In [17], a new hidden node is chosen from a set of randomly generated candidates and the choice is based on the degree of correlation of the basis function output to the residual. As the model selection method in [17], $C_p$ [15] has been applied. In [16], several new hidden nodes are chosen from a set of randomly generated candidates and the choice is based on the degree of reduction of the residual sum of squares; i.e., this method is based on the least squares method. As the model selection method in [16], the final prediction error (FPE) [1] is applied in forward selection and PRESS is used in the backward elimination for candidates selected by FPE; i.e., this is

Manuscript received March 29, 2013.
Manuscript revised September 4, 2013.

[†]The author is with the Faculty of Education, Mie University, Tsu-shi, 514–8507 Japan.
a) E-mail: hagi@edu.mie-u.ac.jp

DOI: 10.1587/transinf.E97.D.98

a hybrid method.

In this paper, we consider a model selection method in RFS. We first clarify the structure of the problem of model selection for the greedy training procedure in RFS. We then give a model selection strategy that is a thresholding method and yields a sparse representation as a result. The model selection strategy is practically implemented by incorporating with the LOOCV method. The greedy procedure considered in this paper and in [18] is essentially the same as the algorithm in [16] except in terms of the introduction of a regularizer and the number of basis functions added in each step. [27] analyzed a condition for the consistency of feature selection for the greedy algorithm that is slightly different from the procedure considered here. A practical model selection method may not be directly obtained by the result of [27], especially in the context of a nonparametric regression problem. In a similar work to this paper, [26], boosting for ridge regression methods is considered including practical model selection, but does not give a sparse representation.

This paper is organized as follows. In Sect. 2, we formulate a greedy procedure with some basic notations. In Sect. 3, we present a model selection method for the greedy procedure defined in Sect. 2. In Sect. 4, we present some numerical examples including real benchmark datasets in which we compare our method with other methods in terms of generalization performances and degrees of sparseness. Finally, in Sect. 5, we discuss the conclusions and future works.

## 2. Formulation and Algorithms

### 2.1 Problem Formulation

Let $\{(\boldsymbol{x}_i, y_i) : i = 1, \dots, n, \boldsymbol{x}_i = (x_{i1}, \dots, x_{id}) \in \mathbb{R}^d, y_i \in \mathbb{R}\}$ be a set of pairs of $d$-dimensional input and one dimensional output training observations. We assume that $y_1, \dots, y_n$ are generated by a rule:

$$y_i = h(\boldsymbol{x}_i) + e_i,\ i = 1, \dots, n, \tag{1}$$

where $h$ is a target function on $\mathbb{R}^d$ and $e_1, \dots, e_n$ are i.i.d. observations from a probability distribution with mean 0 and variance $\sigma^2$. To use matrix notations, we define $\boldsymbol{y} = (y_1, \dots, y_n)'$, $\boldsymbol{e} = (e_1, \dots, e_n)'$, and $\boldsymbol{h} = (h(\boldsymbol{x}_1), \dots, h(\boldsymbol{x}_n))'$, where $'$ stands for the matrix transpose.

We consider a regression method based on a linear sum of an $m$-subset of a set of $n$ functions on $\mathbb{R}^d$. The $n$ functions are denoted by $g_1, \dots, g_n$, which we call basis functions. We define $\boldsymbol{g}_j = (g_j(\boldsymbol{x}_1), \dots, g_j(\boldsymbol{x}_n))'$. Throughout this paper, we assume that $\boldsymbol{g}_1, \dots, \boldsymbol{g}_n$ are linearly independent. We define $N = \{1, \dots, n\}$. Let $\boldsymbol{s} = (l_1, \dots, l_s)$ be an $s$-dimensional index vector whose elements are in $N$. We then define $G_{\boldsymbol{s}} = (\boldsymbol{g}_{l_1}, \dots, \boldsymbol{g}_{l_s})$, which is $n \times s$ matrix.

When $m$ is small relative to $n$, we attempt to obtain a sparse representation of a target function in a nonparametric regression problem. Throughout this paper, we assume that $m < n$. For a fixed $m$, we define $\boldsymbol{\alpha}_m = (\alpha_1, \dots, \alpha_m)$, where $\alpha_j \in N$ for $j = 1, \dots, m$ and $\alpha_1 \neq \cdots \neq \alpha_m$. We also use $\boldsymbol{\alpha}_k$ as a subset of $N$ if there are no confusions; i.e., $\boldsymbol{\alpha}_k = \{\alpha_1, \dots, \alpha_k\}$. We then define the above-mentioned function by

$$f_{\boldsymbol{\alpha}_m, \boldsymbol{w}_m}(\boldsymbol{x}) = \sum_{j=1}^{m} w_j g_{\alpha_j}(\boldsymbol{x}),\ \boldsymbol{x} \in \mathbb{R}^d, \tag{2}$$

where $\boldsymbol{w}_m = (w_1, \dots, w_m)'$ is a coefficient vector. We define a cost function by

$$C(\boldsymbol{\alpha}_m, \boldsymbol{w}_m, \lambda) = \|\boldsymbol{y} - G_{\boldsymbol{\alpha}_m} \boldsymbol{w}_m\|^2 + \lambda \|\boldsymbol{w}_m\|^2, \tag{3}$$

where $\lambda \geq 0$ is a regularization parameter introduced to smooth outputs and to stabilize the training process. Under a fixed $m$ and $\lambda$, we estimate $\boldsymbol{\alpha}_m$ and $\boldsymbol{w}_m$ by minimizing $C(\boldsymbol{\alpha}_m, \boldsymbol{w}_m, \lambda)$. Note that this reduces the least squares estimation when $\lambda = 0$. To do this, we first minimize $C(\boldsymbol{\alpha}_m, \boldsymbol{w}_m, \lambda)$ in terms of $\boldsymbol{w}_m$ at each fixed $\boldsymbol{\alpha}_m$. We define $F_{\boldsymbol{\alpha}_m} = G'_{\boldsymbol{\alpha}_m} G_{\boldsymbol{\alpha}_m} + \lambda I_m$, where $I_m$ is the $m \times m$ identity matrix. We also define $H_{\boldsymbol{\alpha}_m} = G_{\boldsymbol{\alpha}_m} F^{-1}_{\boldsymbol{\alpha}_m} G'_{\boldsymbol{\alpha}_m}$ and $P_{\boldsymbol{\alpha}_m} = I_n - H_{\boldsymbol{\alpha}_m}$. If $\lambda = 0$, then $H_{\boldsymbol{\alpha}_m}$ and $P_{\boldsymbol{\alpha}_m}$ are usually called the hat matrix and the residual matrix, respectively, and these are known to be symmetric and idempotent (see, e.g., [19]). As is well known, we then have

$$\widehat{\boldsymbol{w}}_m(\boldsymbol{\alpha}_m) = F^{-1}_{\boldsymbol{\alpha}_m} G'_{\boldsymbol{\alpha}_m} \boldsymbol{y} \tag{4}$$

as the minimizing weight vector of (3) at a fixed $\boldsymbol{\alpha}_m$. Since $\widehat{\boldsymbol{w}}_m(\boldsymbol{\alpha}_m)$ depends on $\boldsymbol{\alpha}_m$, we need to calculate $C(\boldsymbol{\alpha}_m, \widehat{\boldsymbol{w}}(\boldsymbol{\alpha}_m), \lambda)$ for all choices of $\boldsymbol{\alpha}_m$ in which the number of choices is $\binom{n}{m}$. We then minimize $C(\boldsymbol{\alpha}_m, \widehat{\boldsymbol{w}}(\boldsymbol{\alpha}_m), \lambda)$ with respect to $\boldsymbol{\alpha}_m$. However, this procedure is computationally expensive. To alleviate this problem, a greedy strategy is usually employed.

### 2.2 A Greedy Algorithm

A greedy algorithm for reducing $C(\boldsymbol{\alpha}_m, \boldsymbol{w}_m, \lambda)$ is as follows (see also [17], [18]). Set $\widehat{\boldsymbol{\alpha}}_0 = \{\}$, $N_0 = \{1, \dots, n\}$, and $P_0 = I_n$. Repeat the following procedure for $k = 1, \dots, \bar{k}$, where $\bar{k}$ is chosen by users.

(1) Calculate

$$c^2_{k,l} = \boldsymbol{y}' H_{k,l} \boldsymbol{y} \tag{5}$$

for $l \in N_{k-1}$, where

$$H_{k,l} = \frac{P_{k-1} \boldsymbol{g}_l \boldsymbol{g}'_l P_{k-1}}{\lambda + \boldsymbol{g}'_l P_{k-1} \boldsymbol{g}_l}. \tag{6}$$

(2) Find $\widehat{\alpha}_k = \arg\max_{l \in N_{k-1}} c^2_{k,l}$.

(3) Set $\widehat{\boldsymbol{\alpha}}_k = \widehat{\boldsymbol{\alpha}}_{k-1} \bigcup \{\widehat{\alpha}_k\}$, $N_k = N_{k-1} \backslash \{\widehat{\alpha}_k\}$, and

$$P_k = P_{k-1} - H_{k, \widehat{\alpha}_k}, \tag{7}$$

where \ indicates the subtraction of sets.

We define $\widehat{\boldsymbol{\alpha}}_{k,l} = (\widehat{\alpha}_1, \dots, \widehat{\alpha}_{k-1}, l)$. The weight vector $\boldsymbol{w} = \boldsymbol{w}(\widehat{\boldsymbol{\alpha}}_{k,l})$ that minimizes $C(\boldsymbol{\alpha}, \boldsymbol{w}, \lambda)$ at $\boldsymbol{\alpha} = \widehat{\boldsymbol{\alpha}}_{k,l}$ is given by

$$\widehat{\boldsymbol{w}}(\widehat{\alpha}_{k,l}) = F^{-1}_{\widehat{\alpha}_{k,l}} G'_{\widehat{\alpha}_{k,l}} \boldsymbol{y}. \tag{8}$$

As shown in Appendix A,

$$C(\widehat{\boldsymbol{\alpha}}_{k-1}, \widehat{\boldsymbol{w}}(\widehat{\boldsymbol{\alpha}}_{k-1}), \lambda) = C(\widehat{\boldsymbol{\alpha}}_{k,l}, \widehat{\boldsymbol{w}}(\widehat{\boldsymbol{\alpha}}_{k,l}), \lambda) - c^2_{k,l} \tag{9}$$

holds (see also [17], [18]). Thus, the above procedure minimizes the entire cost function for $k$ basis functions under a fixed $\widehat{\boldsymbol{\alpha}}_{k-1}$ selected up to the previous step. Note that $c^2_{k,l}$ is calculated for a newly appended basis function and represents a reduction of the cost function due to the appended basis function. If $\lambda = 0$, then it is a reduction of a residual sum of squares. It is easy to see that $P_k = P_{\widehat{\alpha}_k}$ by (A·2). By using $P_k$, we can calculate residuals at the $k$-th step by

$$\boldsymbol{r}_k = P_k \boldsymbol{y}. \tag{10}$$

A greedy algorithm considered here does not need a calculation of inverse of matrix in each step. However, the number of multiplications increases in proportion to at most $n^2$. The number of comparisons to find a best fit basis function also increases as $n$ increases basically. These calculations are repeated $\bar{k}$ times that is the number of greedy steps, or equivalently, the number of basis functions that are selected as candidates for model selection. Therefore, it takes much time when $\bar{k}$ is large while $\bar{k}$ should be sufficiently large for a better fitting.

### 2.3 Relation to OLS

We consider the following step (1') instead of (1) in the above procedure.

(1') If $k = 1$, we set $\boldsymbol{q}_{1,l} = \boldsymbol{g}_l$ for $l \in N_0$. If $k \geq 2$, we calculate

$$\boldsymbol{q}_{k,l} = \boldsymbol{q}_{k-1,l} - \beta_{l,k}\boldsymbol{q}_{k-1,\widehat{\alpha}_{k-1}} \tag{11}$$

for $l \in N_{k-1}$, where

$$\beta_{l,k} = \frac{\boldsymbol{q}'_{k-1,l}\boldsymbol{q}_{k-1,\widehat{\alpha}_{k-1}}}{\|\boldsymbol{q}_{k-1,\widehat{\alpha}_{k-1}}\|^2}. \tag{12}$$

For $l \in N_{k-1}$, we then calculate $c^2_{k,l}$ with

$$H_{k,l} = \frac{P_{k-1}\boldsymbol{q}_{k,l}\boldsymbol{q}'_{k,l}P_{k-1}}{\lambda + \boldsymbol{q}'_{k,l}P_{k-1}\boldsymbol{q}_{k,l}}. \tag{13}$$

We simply write $\boldsymbol{q}_{\widehat{\alpha}_k}$ instead of $\boldsymbol{q}_{k,\widehat{\alpha}_k}$. By induction, using (11), we can see that $\{\boldsymbol{q}_{\widehat{\alpha}_1}, \ldots, \boldsymbol{q}_{\widehat{\alpha}_{k-1}}, \boldsymbol{q}_{k,l}\}$ is a set of orthogonal vectors for any $l \in N_{k-1}$. Note that $\boldsymbol{q}_{k,l}$, $l \in N_{k-1}$ are not necessarily orthogonal. By applying (11) recursively, we also obtain

$$\boldsymbol{q}_{k,l} = \boldsymbol{g}_l - \sum_{j=1}^{k-1}\beta_{l,j}\boldsymbol{q}_{j,\widehat{\alpha}_j}. \tag{14}$$

We define $\widehat{\boldsymbol{\alpha}}_{k,l} = (\widehat{\alpha}_1, \ldots, \widehat{\alpha}_{k-1}, l)$ and $Q_{\widehat{\alpha}_{k,l}} = (\boldsymbol{q}_{\widehat{\alpha}_1}, \ldots, \boldsymbol{q}_{\widehat{\alpha}_{k-1}}, \boldsymbol{q}_{k,l})$ which becomes an $n \times k$ orthogonal matrix. We define a cost function given by

$$C(\boldsymbol{\alpha}_k, \boldsymbol{v}_k, \lambda) = \|\boldsymbol{y} - Q_{\alpha_k}\boldsymbol{v}_k\|^2 + \lambda\boldsymbol{v}'_k\boldsymbol{v}_k, \tag{15}$$

where $\boldsymbol{v}_k \in \mathbb{R}^k$ is a coefficient vector of the orthogonalized vectors. The minimizing coefficient vector of the cost function at $\widehat{\boldsymbol{\alpha}}_{k,l}$ is given by

$$\widehat{\boldsymbol{v}}_k(\widehat{\boldsymbol{\alpha}}_{k,l}) = (Q'_{\widehat{\alpha}_{k,l}}Q_{\widehat{\alpha}_{k,l}} + \lambda I_n)^{-1}Q_{\widehat{\alpha}_{k,l}}\boldsymbol{y}. \tag{16}$$

By replacing $(G_{\widehat{\alpha}_{k,l}}, \boldsymbol{w}(\widehat{\boldsymbol{\alpha}}_{k,l}))$ with $(Q_{\widehat{\alpha}_{k,l}}, \boldsymbol{v}(\widehat{\boldsymbol{\alpha}}_{k,l}))$ in Appendix A, we obtain

$$C(\widehat{\boldsymbol{\alpha}}_{k,l}, \boldsymbol{v}_k(\widehat{\boldsymbol{\alpha}}_{k,l}), \lambda) = C(\widehat{\boldsymbol{\alpha}}_{k-1}, \boldsymbol{v}_k(\widehat{\boldsymbol{\alpha}}_{k-1}), \lambda) - c^2_{k,l}, \tag{17}$$

where $c_{k,l}$ is defined in step (1'). The algorithm using (1') instead of (1) is OLS under the regularized cost function. Since $\boldsymbol{q}_{\widehat{\alpha}_1}, \ldots, \boldsymbol{q}_{\widehat{\alpha}_{k-1}}, \boldsymbol{q}_l$ are orthogonal, it is easy to see that $P_{k-1}\boldsymbol{q}_{k,l} = \boldsymbol{q}_{k,l}$ and $c^2_{k,l} = (\boldsymbol{q}_{k,l}\boldsymbol{y})^2/(\lambda + \|\boldsymbol{q}_{k,l}\|^2)$ hold. This implies that we only need vector calculations to obtain $c^2_{k,l}$. Note that the basis function selected at each step can be different for the naive greedy algorithm and OLS since OLS selects an orthogonalized basis function at each step.

## 3. Model Selection for the Greedy Procedure

We here assume that $\boldsymbol{e} \sim N(\boldsymbol{0}, \sigma^2 I_n)$; i.e., $e_1, \ldots, e_n$ are i.i.d. observations from a normal distribution with mean 0 and variance $\sigma^2$. We also assume that $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ are non-stochastic for a while.

### 3.1 Properties of $c^2_{k,\widehat{\alpha}_k}$

We define $a_{k,l} = \lambda + \boldsymbol{g}'_l P_{k-1}\boldsymbol{g}_l$. If we define

$$Z_{k,l} = \boldsymbol{g}'_l P_{k-1}\boldsymbol{y} / \sqrt{a_{k,l}}, \tag{18}$$

we then have $c^2_{k,l} = Z^2_{k,l}$. $Z_{k,l}$ exists for $\lambda \geq 0$ since $\boldsymbol{g}'_l P_{k-1}\boldsymbol{g}_l > 0$ by (A·8) in Appendix B. Note that in a practical computation, this may not hold when $\lambda = 0$ because a numerical instability problem arises. By the definition of $\boldsymbol{y}$, we further have

$$Z_{k,l} = \delta_{k,l} + \xi_{k,l} \tag{19}$$

for $l \in N_{k-1}$, where

$$\delta_{k,l} = (\boldsymbol{h}' P_{\widehat{\alpha}_{k-1}}\boldsymbol{g}_l) / \sqrt{a_{k,l}}, \tag{20}$$

$$\xi_{k,l} = (\boldsymbol{e}' P_{\widehat{\alpha}_{k-1}}\boldsymbol{g}_l) / \sqrt{a_{k,l}}. \tag{21}$$

In (19), $\delta_{k,l}$ relates to a target function and $\xi_{k,l}$ relates to additive noise. Thus, (19) represents the bias and variance decomposition of a reduction of the cost function. Let $\boldsymbol{l} = (l_1, \ldots, l_{n_k})$ be an index vector constructed by enumerating all the elements of $N_{k-1}$ in ascending order, where $n_k = n - (k-1)$. We define $\boldsymbol{Z}_{k,\boldsymbol{l}} = (Z_{k,l_1}, \ldots, Z_{k,l_{n_k}})'$, $\boldsymbol{\delta}_{k,\boldsymbol{l}} = (\delta_{k,l_1}, \ldots, \delta_{k,l_{n_k}})'$, $\boldsymbol{\xi}_{k,\boldsymbol{l}} = (\xi_{k,l_1}, \ldots, \xi_{k,l_{n_k}})'$, and $A_{k,\boldsymbol{l}} = \mathrm{diag}(a_{k,l_1}, \ldots, a_{k,l_{n_k}})$.

Since $\boldsymbol{\xi}'_{k,\boldsymbol{l}} = \boldsymbol{e}' P_{k-1} G_{\boldsymbol{l}} A^{-1/2}_{k,\boldsymbol{l}}$ holds, it is easy to see that the conditional expectation and covariance matrix of $\boldsymbol{\xi}_{k,\boldsymbol{l}}$ given $\widehat{\boldsymbol{\alpha}}_{k-1} = \boldsymbol{\alpha}_{k-1}$ are

$$\mathbb{E}\{\boldsymbol{\xi}_{k,\boldsymbol{l}}|\widehat{\boldsymbol{\alpha}}_{k-1}=\boldsymbol{\alpha}_{k-1}\}=\mathbf{0}_{n_k}, \tag{22}$$

$$\mathbb{E}\{\boldsymbol{\xi}_{k,\boldsymbol{l}}\boldsymbol{\xi}'_{k,\boldsymbol{l}}|\widehat{\boldsymbol{\alpha}}_{k-1}=\boldsymbol{\alpha}_{k-1}\}=\sigma^2 A_{k,\boldsymbol{l}}^{-1/2}G'_{\boldsymbol{l}}P^2_{\widehat{\boldsymbol{\alpha}}_{k-1}}G_{\boldsymbol{l}}A_{k,\boldsymbol{l}}^{-1/2} \tag{23}$$

by the property of $\boldsymbol{e}$, where $\mathbf{0}_{n_k}$ is the $n_k$-dimensional zero vector. $\boldsymbol{\xi}_{k,\boldsymbol{l}}$ is a linear transformation of $\boldsymbol{e}$. Thus, by p.31, [4], the joint probability distribution of $\boldsymbol{\xi}_{k,\boldsymbol{l}}$ given $\widehat{\boldsymbol{\alpha}}_{k-1}=\boldsymbol{\alpha}_{k-1}$ is the normal distribution that is not possibly degenerate. Therefore, for $l\in N_{k-1}$, the conditional probability distribution of $\xi_{k,l}$ given $\widehat{\boldsymbol{\alpha}}_{k-1}=\boldsymbol{\alpha}_{k-1}$ is a normal distribution with zero mean and variance:

$$\sigma_l^2=\sigma^2(\boldsymbol{g}'_l P^2_{\widehat{\boldsymbol{\alpha}}_{k-1}}\boldsymbol{g}_l)/a_{k,l}. \tag{24}$$

We have $\sigma_l^2>0$ for $\lambda\geq 0$ by (A· 8) and (A· 9) in Appendix B. By the definition of $a_{k,l}$ and (A· 11) in Appendix B, we also have

$$\sigma_l^2\leq\sigma^2 \tag{25}$$

for any $l\in N_{k-1}$ and $\lambda\geq 0$.

We define $\overline{\delta}_k=\max_{l\in N_{k-1}}|\delta_{k,l}|$ and $\theta_{k,n}(\epsilon)=\sqrt{(2+\epsilon)\log n_k}$, where $n_k=n-(k-1)$ and $\epsilon>0$ in which $n$ is the number of data and $k-1$ is the number of selected basis functions up to the $k$th step. By using the well-known upper-tail inequality for a normal distribution and (25), we have

$$\begin{aligned}
&\mathbb{P}\left\{|Z_{k,\widehat{\alpha}_k}|>\overline{\delta}_{k,l}+\sigma\theta_{k,n}(\epsilon)|\widehat{\boldsymbol{\alpha}}_{k-1}=\boldsymbol{\alpha}_{k-1}\right\}\\
&=\mathbb{P}\left\{\max_{l\in N_{k-1}}|Z_{k,l}|>\overline{\delta}_{k,l}+\sigma\theta_{k,n}(\epsilon)|\widehat{\boldsymbol{\alpha}}_{k-1}=\boldsymbol{\alpha}_{k-1}\right\}\\
&=\mathbb{P}\left\{\bigcup_{l\in N_{k-1}}\left\{|Z_{k,l}|>\overline{\delta}_{k,l}+\sigma\theta_{k,n}(\epsilon)\right\}|\widehat{\boldsymbol{\alpha}}_{k-1}=\boldsymbol{\alpha}_{k-1}\right\}\\
&\leq\mathbb{P}\left\{\bigcup_{l\in N_{k-1}}\{|\xi_{k,l}|>\sigma\theta_{k,n}(\epsilon)\}|\widehat{\boldsymbol{\alpha}}_{k-1}=\boldsymbol{\alpha}_{k-1}\right\}\\
&\leq\sum_{l\in N_{k-1}}\mathbb{P}\left\{|\xi_{k,l}/\sigma_l|>\frac{\sigma}{\sigma_l}\theta_{k,n}(\epsilon)|\widehat{\boldsymbol{\alpha}}_{k-1}=\boldsymbol{\alpha}_{k-1}\right\}\\
&\leq\frac{n-(k-1)}{\theta_{k,n}(\epsilon)\sqrt{2\pi}}\exp\{-\theta_{k,n}(\epsilon)^2/2\}\\
&=\frac{C_\epsilon}{n_k^{\epsilon/2}\sqrt{\log n_k}},
\end{aligned} \tag{26}$$

where $C_\epsilon$ is a positive constant. We thus have

$$\begin{aligned}
&\mathbb{P}\left\{|c_{k,\widehat{\alpha}_k}|>\overline{\delta}_k+\sigma\theta_{k,n}(\epsilon)\right\}\\
&=\sum_{\boldsymbol{\alpha}_{k-1}}\mathbb{P}\left\{|c_{k,\widehat{\alpha}_k}|>\overline{\delta}_k+\sigma\theta_{k,n}(\epsilon)|\widehat{\boldsymbol{\alpha}}_{k-1}=\boldsymbol{\alpha}_{k-1}\right\}\mathbb{P}\left\{\widehat{\boldsymbol{\alpha}}_{k-1}=\boldsymbol{\alpha}_{k-1}\right\}\\
&\leq\frac{C_\epsilon}{n_k^{\epsilon/2}\sqrt{\log n_k}},
\end{aligned} \tag{27}$$

where $\sum_{\boldsymbol{\alpha}_{k-1}}$ denotes the sum for $\binom{n}{k-1}$ choices of $\boldsymbol{\alpha}_{k-1}$. When $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n$ are stochastic, this bound is also valid if $\boldsymbol{g}_1,\ldots,\boldsymbol{g}_n$ are linearly independent with probability one. In this case, $\overline{\delta}_k$ should be a bound under any choice of $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n$.

If $n$ is sufficiently larger than $k$, (27) can be very small. Therefore, $\overline{\delta}_k+\sigma\theta_{k,n}(\epsilon)$ is an upper bound for the cost reduction obtained by adding a new basis function in which the first and second terms correspond to the bias and variance bounds, respectively. When $\lambda=0$, it is an upper bound for the residual reduction. If a target function has a truly sparse representation in terms of the assumed basis functions and all of the true basis functions have already been selected at the $(k-1)$-th step, then $\delta_{k,l}=0$ for all $l\in N_{k-1}$. Unfortunately, this may not be feasible in practical applications of nonparametric regression methods. Even when this is not exactly satisfied, we can expect $\delta_{k,l}$ to be sufficiently small at an appropriate $k$ since the greedy procedure may be able to effectively reduce a bias. This is a significant point for applying a greedy procedure. In other words, at the $k$, a variance in a cost reduction may be dominated. However, it is bounded by $\sigma\theta_{k,n}(\epsilon)$. We therefore consider the $k$-th basis function as irrelevant to a target function and only relevant to noise when

$$|\widehat{c}_{k,\widehat{\alpha}_k}|\leq\sigma\theta_{k,n}(\epsilon) \tag{28}$$

holds.

### 3.2 Greedy Algorithm for a Special Case

In the previous subsection, the important point is the structure of the problem for deriving the bound; i.e., we need to evaluate the maximum value among certain random variables. This comes from an intrinsic property of the greedy procedure. In this subsection, we will see that this structure is important in considering the model selection problem of the greedy procedure.

For the greedy algorithm, we bound $\xi^2_{k,\widehat{\alpha}_k}=\max_{l\in N_{k-1}}\xi^2_{k,l}$ to evaluate the variance in cost reduction. This is required because it selects a basis function that maximally reduces the defined cost. If we fix or randomly select a new basis function, it is easy to see that the expectation of $\xi^2_{k,l}$ is less than $\sigma^2$, and equal to $\sigma^2$ when $\lambda=0$. Indeed, this holds even when an additive noise does not have a normal distribution. This fact plays an essential role in deriving FPE and $C_p$. This is true even in AIC; i.e., it is based on the fact that $\xi^2_{k,l}$ has a $\chi^2$ distribution with one degree of freedom as the asymptotic distribution when $\lambda=0$. The penalty terms of these criteria play the role of correcting the residual reduction for noise in estimating a generalization error based on a training error. In other words, these model selection criteria estimate residual or cost reduction as a constant that does not depend on $n$. In the greedy algorithm, if the size of $N_{k-1}$ increases, $\xi^2_{k,\widehat{\alpha}_k}$ may increase basically since the number of candidates increases. This implies that $\xi^2_{k,\widehat{\alpha}_k}$ may be an increasing sequence of $n$ if $k$ is fixed because the number of basis functions increases as $n$ increases in our setting of a nonparametric regression problem. Indeed, if $\boldsymbol{g}_1,\ldots,\boldsymbol{g}_n$ are orthogonal, we can show that for a fixed $k$, the probability of the event $\{|\xi_{k,\widehat{\alpha}_k}|>\sigma\theta_{k,n}(-\epsilon)\}$ with $\epsilon>0$ goes to zero as $n$ goes to $\infty$.

To see this, we assume that $\boldsymbol{g}_1, \ldots, \boldsymbol{g}_n$ are orthogonal. For simplicity, we assume that $\lambda = 0$ holds here. By recalling that $P_{\widehat{\boldsymbol{\alpha}}_{k-1}} = I_n - H_{\widehat{\boldsymbol{\alpha}}_{k-1}}$ and $H_{\widehat{\boldsymbol{\alpha}}_{k-1}} = G_{\widehat{\boldsymbol{\alpha}}_{k-1}} F^{-1}_{\widehat{\boldsymbol{\alpha}}_{k-1}} G'_{\widehat{\boldsymbol{\alpha}}_{k-1}}$, we have $P_{\widehat{\boldsymbol{\alpha}}_{k-1}} \boldsymbol{g}_l = \boldsymbol{g}_l$ for $l \in N_{k-1}$ since $l \notin \widehat{\boldsymbol{\alpha}}_{k-1}$. We thus have $\mathbb{E}\{\boldsymbol{\xi}_{k,\boldsymbol{l}} \boldsymbol{\xi}'_{k,\boldsymbol{l}} | \widehat{\boldsymbol{\alpha}}_{k-1} = \boldsymbol{\alpha}_{k-1}\} = \sigma^2 I_n$ by (23) and the definition of $A_{k,\boldsymbol{l}}$. This implies that $\xi^2_{k,l}$, $l \in N_{k-1}$ are i.i.d. samples from a $\chi^2$ distribution with one degree of freedom. By Appendix C, for $\epsilon > 0$, we therefore have

$$1 - \mathbb{P}\left\{ \max_{l \in N_{k-1}} |\xi_{k,l}| > \sigma\theta_{k,n}(-\epsilon) | \widehat{\boldsymbol{\alpha}}_{k-1} = \boldsymbol{\alpha}_{k-1} \right\} \leq \frac{\sqrt{n^{-\epsilon} \log n}}{2C_\epsilon}, \tag{29}$$

where $C_\epsilon$ is a positive constant. This implies that the probability that a variance in a residual reduction is bounded below by $C \log n$ for a positive constant $C$ can be high when $n$ is large. In other words, $\xi^2_{k,\widehat{\alpha}_k}$ is really an increasing function of $n$ when $\boldsymbol{g}_1, \ldots, \boldsymbol{g}_n$ are orthogonal. Therefore, FPE, $C_p$, and AIC may underestimate a residual reduction and tend to select a larger number of basis functions than needed to represent a target function. If $\boldsymbol{g}_1, \ldots, \boldsymbol{g}_n$ are not orthogonal, then $\xi^2_{k,\widehat{\alpha}_k}$ can be small depending on the correlations among $\xi_{k,l}$ [14]. This correlation structure is determined by the degree of linear dependency among $\boldsymbol{g}_1, \ldots, \boldsymbol{g}_n$. In an extreme case, if all of the basis functions are linearly dependent, then we prepare only one basis function and fit the given data by that basis function. In an alternative extreme case, as seen here, a residual reduction is lower bounded by $C \log n$ if all basis functions are orthogonal. It is natural that $|\xi_{k,\widehat{\alpha}_k}|$ is an increasing function of $n$ with a high probability when the number of linear independent basis functions increases with $n$. Actually, in a greedy version of the extreme learning machine, [16] first chooses candidates of basis functions by FPE in the forward selection while using PRESS for backward elimination. The requirement of the backward elimination may constitute empirical evidence for the above discussion. Although the exact evaluation is left to future study, the upper bound (27) is valid in any case and is expected to be useful in practical applications.

### 3.3 Implementation Issue

We discuss here the implementation of a model selection method based on (28), in which we attempt to incorporate the above result with the LOOCV method.

We first summarize a model selection method based on the LOOCV error. In implementing the model selection method, we set $\overline{k}$, the number of candidates for a model selection. $\overline{k}$ should be sufficiently large, but small enough to keep computational time reasonable. Let LCVE$(k)$ be a LOOCV error at $k$ in $\{1, \ldots, \overline{k}\}$ given by

$$\mathrm{LCVE}(k) = \sum_{i=1}^{n} \left( \frac{r_i}{1 - h_{ii}} \right)^2, \tag{30}$$

where $r_i$ is a residual for the $i$-th observations and $h_{ii}$ is the $i$-th diagonal element of $H_{\widehat{\alpha}_k}$. At each $k$, these are calculated by using $P_k$, in which we can use (10) and $H_{\widehat{\alpha}_k} = I_n - P_k$. We find $\widehat{k}_{\mathrm{LCV}} = \arg\min_{1 \leq k \leq \overline{k}} \mathrm{LCVE}(k)$, which is the selected number of basis functions according to the LOOCV error. We refer to this method as LOOCV. Note that if $\widehat{k}_{\mathrm{LCV}} = \overline{k}$, we may need to set a larger $\overline{k}$.

We define $K = \{k \mid 1 \leq k \leq \widehat{k}_{\mathrm{LCV}}, \ |c_{k,\widehat{\alpha}_k}| \geq \sigma\theta_{k,n}(\epsilon)\}$; i.e., a set of indices of basis functions for which the cost reduction is larger than $\sigma\theta_{k,n}(\epsilon)$. We consider to select $K$ as indices of significant basis functions, which is a thresholding on the cost reduction. In this thresholding method, if we happen to remove the basis functions that represent a target function, a large bias arises. There is a possibility of this elimination because $c_{k,l}$, $l \in N_{k-1}$ have a stochastic nature and the obtained threshold level is an upper bound for the variance. To reduce the accidental elimination and for a safety purpose in the implementation, we find $\widehat{k} = \max_{j \in K} j$ and select $\{1, \ldots, \widehat{k}\}$ as indices of significant basis functions. In other words, $|c_{k,\widehat{\alpha}_k}| < \sigma\theta_{k,n}(\epsilon)$ holds for any $k$ that satisfy $\widehat{k} < k \leq \widehat{k}_{\mathrm{LCV}}$ if $\widehat{k} < \widehat{k}_{\mathrm{LCV}}$. Since we choose basis functions for the candidates chosen by LOOCV, this method yields a more sparse representation than the representation given by LOOCV; i.e., $\widehat{k} \leq \widehat{k}_{\mathrm{LCV}}$. We refer to this method as thresholding for a cost reduction (TCR) and denote $\widehat{k}$ by $\widehat{k}_{\mathrm{TCR}}$. In implementing TCR, we need to choose $\epsilon$ and $\sigma^2$. Since $\epsilon$ is an arbitrary small value, we set $\epsilon = 0$. Note that (27) goes to zero as $n$ goes to $\infty$ even when $\epsilon = 0$. As an estimate of $\sigma^2$, we consider employing LCVE$(\widehat{k}_{\mathrm{LCV}})$. This scheme is effectively combined with a hyperparameter selection based on the LOOCV error.

The remainder of the implementation of TCR is the choice of a regularization parameter. Generally, the regularization parameter is introduced to smooth the output and/or to stabilizing the training process. The former purpose is needed for better generalization performance of the estimated model. However, estimation using TCR is expected to provide a model with better generalization performance by restricting the complexity of the model; i.e., the number of basis functions. On the other hand, the calculation of $H_{k,l}$ in the greedy procedure can be numerically unstable if $\lambda = 0$; e.g., $P_{k-1}\boldsymbol{g}_l$ can be very small if $\boldsymbol{g}_l$ is nearly linearly dependent with $\boldsymbol{g}_{\widehat{\alpha}_j}$, $j = 1, \ldots, k-1$. Therefore, we may need the regularization parameter to avoid this instability, but we may not need it to maintain a better generalization performance. We thus set an appropriate small value for the regularization parameter in the implementation of TCR. Note that TCR can be useful even when we can choose a value for the regularization parameter that realizes appropriate smoothing. However, choosing an optimal value for the regularization parameter in addition to the hyperparameter may take much time. We can avoid this problem by using TCR to obtain good generalization performance.

Lastly, we should note that TCR can be also applied to OLS since (28) holds when $\boldsymbol{g}_l$ is replaced with $\boldsymbol{q}_{k,l}$. In this meaning, (28) is a universal bound in terms of basis functions.

## 4. Numerical Experiments

### 4.1 Some Benchmark Examples

In this section, we compare the degree of sparseness and generalization performance of TCR to those of other methods through numerical experiments on real benchmark datasets from [25]. The alternative methods include LOOCV in Sect. 3.3, FPE [1], LASSO [23] and one-standard error rule (OSER) (p.216, [13]) for LOOCV error. For LASSO, we directly use the "glmnet" package of R that is a popular free software for statistical data analysis.

We employ a Gaussian basis function with a single width parameter that is common across all basis functions; i.e. $g_k(\boldsymbol{x}) = \exp(-\|\boldsymbol{x} - \boldsymbol{x}_k\|^2/\tau)$ with the width parameter $\tau > 0$. Samples of each dataset are divided into a training set and a test set. A model is trained for a training set by using each method; i.e. we obtain a set of basis functions and their coefficients by each method. The trained model is tested on a test set. The test error is defined as the mean squared error on the test set. We repeat this procedure for 20 different randomly chosen pairs of sample sets and calculate the averaged test error and averaged number of selected basis functions for 20 trials. The names of the datasets are shown in Table 1, with the numbers of training data, test data, and inputs. For each dataset, values for all variables are normalized to zero mean and unit variance.

The set of candidate values for the hyperparameter is shown in the last column of Table 1. These are chosen based on the results of pre-simulations. For all methods, the hyperparameter value is chosen from each candidate set based on LOOCV error (30). Except TCR, we choose a regularization parameter value in $\{10^{-6}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1\}$. In TCR, we set $\lambda = 10^{-6}$ for all datasets. For the choice of a regularization parameter, the LOOCV error is employed in LOOCV, FPE and OSER methods while 10-fold cross validation error is employed in LASSO to save time. We set $\overline{k} = 200$ (the number of candidates of basis functions for model selection) for "ailerons", "delta_ailerons", "elevators" and "delta_elevators" datasets and $\overline{k} = 50$ for "housing" and "auto" datasets. Note that LOOCV and FPE methods select basis functions from $\overline{k}$ candidates according to LOOCV error and FPE criterion respectively. The $\overline{k}$ candidates are selected by the greedy algorithm employed in this paper. OSER applied to LOOCV errors for the $\overline{k}$ candidates under the hyperparameter and regularization parameter that are chosen according to LOOCV error.

In Tables 2 and 3, we show the test errors and the selected numbers of the basis functions respectively. In Tables 2 and 3, we can see that the generalization performance and sparseness of LOOCV are comparable to those of FPE. The selected number of basis functions in OSER is necessarily smaller than that in LOOCV. And, the average test error of OSER is less than that of LOOCV while their difference is within the standard deviations. However, this result may say that the models selected by LOOCV and FPE tend to over-fit to training data. TCR shows a better generalization performance compared to LOOCV, OSER and FPE in average while it may not be notable when we take account of the standard deviations. LASSO shows a better generalization performance compared to TCR except "ailerons" dataset while the difference in test errors may not be notable in some datasets when we take account of the standard deviations. On the other hand, the sparseness of TCR is notable compared to LOOCV, OSER, FPE and LASSO as found in Table 3. Since [16] employs LOOCV for the backward elimination, we can say that TCR can provide a sparser representation than in [16]. In Table 3, we can see that LASSO is not stable in terms of the degree of sparseness while it shows a relatively stable generalization performance; e.g. it is seen typically in "elevators" dataset. In LASSO, the degree of sparseness is controlled by a regularization parameter under a fixed hyperparameter. Therefore, the result tells us that the sparseness may be sensitive to a regularization parameter around an optimal value while the test error may not be so sensitive. In LASSO, to find a reasonable sparse representation, we may need a set of the large number of candidate values for a regularization parameter while it takes much time in estimation.

From this experiment, we can say that TCR provides a model which shows a relatively better generalization performance and has a strong sparseness property. TCR is therefore preferable in terms of the law of parsimony. Note that these facts are true even for the relatively small size datasets; i.e., "housing" and "auto". Moreover, it is important that the preferable property of TCR is realized under a fixed regularization parameter value, by which we are free from a choice of value for a regularization parameter. This is further examined in the next section.

### 4.2 Dependence of Regularization Parameter on TCR

In Fig. 1, we show a relationship between regularization pa-

**Table 1** Names of datasets, the number of training data ($n_{\text{train}}$), test data ($n_{\text{test}}$), inputs ($d$) and the candidate values for a hyperparameter $\tau$.

| names of datasets | $n_{\text{train}}$ | $n_{\text{test}}$ | $d$ | candidates for $\tau$ |
|---|---|---|---|---|
| ailerons | 1000 | 2000 | 41 | $\{120, 140, 160, 180, 200\}$ |
| delta_ailerons | 1000 | 2000 | 6 | $\{2, 5, 10, 15, 20\}$ |
| elevators | 1000 | 2000 | 19 | $\{25, 50, 75, 100, 125\}$ |
| delta_elevators | 1000 | 2000 | 7 | $\{2, 5, 10, 30, 50\}$ |
| housing | 150 | 356 | 14 | $\{20, 30, 40, 50, 60\}$ |
| auto | 150 | 248 | 8 | $\{2, 5, 10, 15, 20\}$ |

**Table 2** Averaged test errors with the standard deviation in the bracket.

| Data Set | LOOCV | OSER | FPE | TCR | LASSO |
|---|---|---|---|---|---|
| ailerons | 0.215(0.021) | 0.2(0.019) | 0.213(0.033) | 0.187(0.012) | 0.19(0.01) |
| delta_ailerons | 0.396(0.041) | 0.381(0.04) | 0.376(0.046) | 0.359(0.04) | 0.331(0.016) |
| elevators | 0.156(0.025) | 0.154(0.025) | 0.149(0.018) | 0.144(0.018) | 0.135 (0.011) |
| delta_elevators | 0.457(0.033) | 0.441(0.031) | 0.445(0.032) | 0.39(0.024) | 0.373(0.016) |
| housing | 0.213(0.036) | 0.205(0.025) | 0.214(0.034) | 0.214(0.041) | 0.213(0.036) |
| auto | 0.175(0.031) | 0.164(0.025) | 0.175(0.027) | 0.163(0.024) | 0.143(0.018) |

**Table 3** Average number of selected basis functions with the standard deviation in the bracket.

| Data Set | LOOCV | OSER | FPE | TCR | LASSO |
|---|---|---|---|---|---|
| ailerons | 126.8(32.3) | 79.2(34.8) | 141.1(49.5) | 25.8(8.5) | 110.8(135.8) |
| delta_ailerons | 106.1(36.8) | 61.9(26.4) | 96.2(46.5) | 21.9(8.5) | 122.2(220.7) |
| elevators | 182.2(14.2) | 141.2(24.8) | 191.4(16.5) | 46.2(12.7) | 809.2(297.3) |
| delta_elevators | 113.3(34.3) | 66(33.3) | 126.7(44.1) | 11.2(6.9) | 49.6(100.5) |
| housing | 39(7.2) | 29.9(6.5) | 43.4(7.5) | 18.4(4.2) | 59.4(46.6) |
| auto | 41.3(7.7) | 27.6(8.8) | 43.6(8.1) | 15.1(7.3) | 33.8(17.6) |

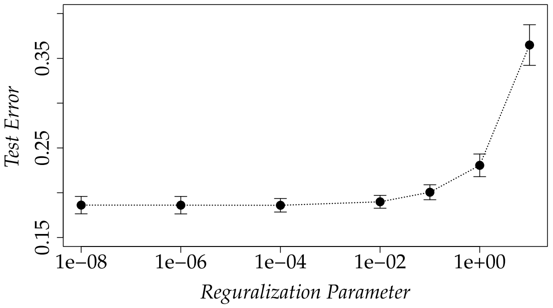

**Fig. 1** Test error of TCR at different values of a regularization parameter. The dataset is "ailerons". The error bar denotes the standard deviation.

rameter, $\lambda$, and test error of TCR for "ailerons" dataset. The number of training and test data are 1000 and 2000 respectively. We set $\bar{k} = 200$. For each fixed regularization parameter, we estimate a model by TCR for a training set and then calculate the test error for a test set. We repeat this procedure 20 times for different choices of data and calculate the mean and standard deviation of the test error. The result is depicted in Fig. 1. The mean test error was minimized at $\lambda = 10^{-4}$ while the minimum test error is almost equal to the test errors at $\lambda = 10^{-8}$ and $\lambda = 10^{-6}$. Therefore, we can say that the generalization performance of a model estimated by TCR is not so sensitive to a pre-determined regularization parameter when it is set to be a small value. This is because a better generalization capability is brought by thresholding; i.e. restricting the number of basis functions. This avoids an increase of test error for a small value of a regularization parameter. On the other hand, the test error is large when the regularization parameter is set to be a large value. Therefore, the choice of a small value for the regularization parameter is sufficient for stability of estimator and a better generalization performance. This makes us free from a choice of value for a regularization parameter.

## 5. Conclusions

In this paper, we proposed a model selection method in regularized forward selection [18]. For the purpose, we focused on the reduction of a cost function, which is brought by appending a new basis function in a greedy training procedure. We first clarified a bias and variance decomposition of the cost reduction and then derived a probabilistic upper bound for the variance of the cost reduction under some conditions. The derived upper bound reflects an essential feature of the greedy training procedure; i.e., it selects a basis function which maximally reduces the cost function. We then proposed a thresholding method for determining significant basis functions by applying the derived upper bound as a threshold level and effectively combining with the leave-one-out cross validation method. The thresholding method with this level identifies whether the appended basis function contributes to overfitting or approximation of a target function. With numerical examples, we verified that we can obtain a sparser representation by using the proposed method than by using a naive LOOCV approach; nevertheless, the generalization performances of both methods were comparable. In the next stage, we need to consider the thresholding method that acts as a stopping criterion in the greedy procedure, which may effectively reduce the overall computational complexity. On the other hand, the well-known forward stagewise and least angle regression [8] are closely related to the forward selection procedure considered here. Those are variations of greedy procedure in which the algorithm basically selects a basis function that maximally reduces residual, or equivalently, maximizes correlation with residual. Therefore, our idea of thresholding here may be applicable to the forward stagewise and least angle regression. This attempt is also a part of our future works. Also, backward elimination is an alternative strategy for building a model. The application of our thresholding method may be effective and possible to reduce a computational cost for model selection in backward elimination. It is also a part of our future works.

### References

[1] H. Akaike, "Fitting autoregressive models for prediction," Annals of the Institue of Statistical Mathematics, vol.21, pp.243–247, 1969.

[2] H. Akaike, "Information theory and an extension of the maximum likelihood principle," 2nd International Symposium on Information Theory, B.N. Petrov and F. Csaki (eds.), pp.267–281, Akadimiai Kiado, Budapest, 1973.

[3] D.M. Allen, "The relationship between variable selection and data augmentation and a method for prediction," Technometrics, vol.16, pp.125–127, 1974.

[4] T.W. Anderson, An introduction to multivariate statistical analysis, Third ed., Wiley, 2003.

[5] S. Chen, X. Hong, C.J. Harris, and P.M. Sharkey, "Sparse modeling using orthogonal forward regression with PRESS statistic and regularization," IEEE Trans. Syst. Man Cybern., vol.34, pp.898–911, 2004.

[6] S. Chen, "Orthogonal-least-squares regression: A unified approach for data modeling," Neurocomputing, vol.72, pp.2679–2681, 2009.

[7] N. Cristianini and J. Shawe-Taylor, An introduction to support vector machines and other kernel-based learning methods, Cambridge University Press, 2000.

[8] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least ange regression," The Annals of Statistics, vol.32, pp.407-499, 2004.

[9] G. Feng, G.-B. Huang, Q. Lin, and R. Gay, "Error minimized extreme learning machine with growth of hidden nodes and incremental learning," IEEE Trans. Neural Netw., vol.20, pp.1352–1357, 2009.

[10] K. Hagiwara, "Model selection with componentwise shrinkage in orthogonal regression," IEICE Trans. Fundamentals, vol.E86-A, no.7, pp.1749–1758, July 2003.

[11] K. Hagiwara, "Nonparametric regression method based on orthogonalization and thresholding," IEICE Trans. Inf. & Syst., vol.E94-D, no.8, pp.1610–1619, Aug. 2011.

[12] D.A. Harville, Matrix algebra from a statistician's perspective, Springer, 1997.

[13] T. Hastie, R. Tibshirani, and J. Friedman, The elements of statistical learning, Springer, 2001.

[14] K. Jogdeo, "A simple proof of an inequality for multivariate normal probabilities of rectangles," The Annals of Mathematical Statistics, vol.41, pp.1357–1359, 1970.

[15] C.L. Mallows, "Some comments on $C_p$," Technometrics, vol.15, pp.661–675, 1973.

[16] Y. Lan, Y.C. Soh, and G.-B. Huang, "Two-stage extreme learning machine for regression," Neurocomputing, vol.73, pp.3028–3038, 2010.

[17] Y. Lan, Y.C. Soh, and G.-B. Huang, "Constructive hidden nodes selection of extreme learning machine for regression," Neurocomputing, vol.73, pp.3191–3199, 2010.

[18] M.J.L. Orr, "Introduction to radial basis function networks," Technical report, Institute for Adaptive and Neural Computation, Edinburgh University, 1996.

[19] N. Ravishanker and D.K. Dey, A first course in linear model theory, Chapman & Hall/CRC, 2002.

[20] R. Rifkin, Everything old is new again: A fresh look at historical approaches in machine learning, Ph.D. thesis, 2002.

[21] M. Stone, "Cross-validatory choice and assessment of statistical predictions (with discussion)," J. R. Statist. Soc. B, vol.36, pp.111–147, 1974.

[22] J.A.K. Suykens, J.D. Brabanter, L. Lukas, and J. Vandewalle, "Weighted least squares support vector machines: Robustness and sparse approximation," Neurocomputing, vol.48, pp.85–105, 2002.

[23] R. Tibshirani, "Regression shrinkage and selection via the lasso," J. R. Statist. Soc. B, vol.58, pp.267–288, 1996.

[24] M.E. Tipping, "Sparse Bayesian learning and the relevance vector machine," J. Machine Learning Research, vol.1, pp.211–244, 2001.

[25] L. Torgo, www.liaad.up.pt/~ltorgo/Regression/DataSets.html, Regression datasets, 2005.

[26] G. Tutz and H. Binder, "Boosting ridge regression," Computational Statistics & Data Analysis, vol.51, pp.6044–6059, 2007.

[27] T. Zhang, "On the consistency of feature selection using greedy least squares regression," J. Machine Learning Research, vol.10, pp.555–568, 2009.

[28] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," Neurocomputing, vol.70, pp.489–501, 2006.

## Appendix A: Derivation of (9)

Since we have $G_{\widehat{\boldsymbol{\alpha}}_{k,l}}\widehat{\boldsymbol{w}}(\widehat{\boldsymbol{\alpha}}_{k,l}) = H_{\widehat{\boldsymbol{\alpha}}_{k,l}}\boldsymbol{y}$,

$$\begin{aligned}
&C(\widehat{\boldsymbol{\alpha}}_{k,l}, \widehat{\boldsymbol{w}}(\widehat{\boldsymbol{\alpha}}_{k,l}), \lambda)\\
&= \|\boldsymbol{y} - H_{\widehat{\boldsymbol{\alpha}}_{k,l}}\boldsymbol{y}\|^2 + \lambda\|\widehat{\boldsymbol{w}}_{\widehat{\boldsymbol{\alpha}}_{k,l}}\|^2\\
&= \|\boldsymbol{y}\|^2 - 2\boldsymbol{y}'H_{\widehat{\boldsymbol{\alpha}}_{k,l}}\boldsymbol{y} + \boldsymbol{y}'H^2_{\widehat{\boldsymbol{\alpha}}_{k,l}}\boldsymbol{y} + \lambda\boldsymbol{y}'G_{\widehat{\boldsymbol{\alpha}}_{k,l}}F^{-1}_{\widehat{\boldsymbol{\alpha}}_{k,l}}F^{-1}_{\widehat{\boldsymbol{\alpha}}_{k,l}}G'_{\widehat{\boldsymbol{\alpha}}_{k,l}}\boldsymbol{y}\\
&= \|\boldsymbol{y}\|^2 - 2\boldsymbol{y}'H_{\widehat{\boldsymbol{\alpha}}_{k,l}}\boldsymbol{y} + \boldsymbol{y}'\left(H^2_{\widehat{\boldsymbol{\alpha}}_{k,l}} + \lambda G_{\widehat{\boldsymbol{\alpha}}_{k,l}}F^{-1}_{\widehat{\boldsymbol{\alpha}}_{k,l}}F^{-1}_{\widehat{\boldsymbol{\alpha}}_{k,l}}G'_{\widehat{\boldsymbol{\alpha}}_{k,l}}\right)\boldsymbol{y}\\
&= \|\boldsymbol{y}\|^2 - 2\boldsymbol{y}'H_{\widehat{\boldsymbol{\alpha}}_{k,l}}\boldsymbol{y} + \boldsymbol{y}'G_{\widehat{\boldsymbol{\alpha}}_{k,l}}F^{-1}_{\widehat{\boldsymbol{\alpha}}_{k,l}}G'_{\widehat{\boldsymbol{\alpha}}_{k,l}}\boldsymbol{y}\\
&= \|\boldsymbol{y}\|^2 - \boldsymbol{y}'H_{\widehat{\boldsymbol{\alpha}}_{k,l}}\boldsymbol{y}, \qquad \text{(A$\cdot$ 1)}
\end{aligned}$$

where we use the definition of $H_{\widehat{\boldsymbol{\alpha}}_{k,l}}$ in the last two lines. As in [16], [18], we can show that

$$H_{\widehat{\boldsymbol{\alpha}}_{k,l}} = H_{\widehat{\boldsymbol{\alpha}}_{k-1}} + H_{k,l} \qquad \text{(A$\cdot$ 2)}$$

where $H_{k,l}$ is defined by (6). This proves (9) since we have

$$C(\widehat{\boldsymbol{\alpha}}_{k-1}, \widehat{\boldsymbol{w}}(\widehat{\boldsymbol{\alpha}}_{k-1}), \lambda) = \|\boldsymbol{y}\|^2 - \boldsymbol{y}'H_{\widehat{\boldsymbol{\alpha}}_{k-1}}\boldsymbol{y} \qquad \text{(A$\cdot$ 3)}$$

by replacing $\widehat{\boldsymbol{\alpha}}_{k,l}$ with $\widehat{\boldsymbol{\alpha}}_{k-1}$ in (A· 1).

## Appendix B: Some Properties of $H_{\widehat{\boldsymbol{\alpha}}_{k-1}}$ and $P_{\widehat{\boldsymbol{\alpha}}_{k-1}}$

We write $H_{\widehat{\boldsymbol{\alpha}}_{k-1}} = H_{k-1}(\lambda)$, $P_{k-1}(\lambda) = P_{\widehat{\boldsymbol{\alpha}}_{k-1}}$ and $F_{\widehat{\boldsymbol{\alpha}}_{k-1}} = F_{k-1}(\lambda)$. We here recall that $P_{k-1}(\lambda) = I_n - H_{k-1}(\lambda)$ by the definition of $P_{\widehat{\boldsymbol{\alpha}}_{k-1}}$. We define $q_{k-1}(\boldsymbol{u}, \lambda) = \boldsymbol{u}'H_{k-1}(\lambda)\boldsymbol{u}$, where $\boldsymbol{u}$ is an $n$-dimensional vector. Let $\gamma_1 \geq \cdots \geq \gamma_n$ be eigenvalues of $H_{k-1}(\lambda)$. Since $H_{k-1}(\lambda)$ is symmetric, there exists an orthogonal matrix $Q$ such that $Q'H_{k-1}(\lambda)Q = \Gamma$ or $Q\Gamma Q' = H_{k-1}(\lambda)$, where $\Gamma = \mathrm{diag}(\gamma_1, \ldots, \gamma_n)$.

We here define $F(\rho) = (G'_{\widehat{\boldsymbol{\alpha}}_{k-1}}G_{\widehat{\boldsymbol{\alpha}}_{k-1}} + \rho I_n)$ and $H(\rho) = G_{\widehat{\boldsymbol{\alpha}}_{k-1}}F(\rho)^{-1}G'_{\widehat{\boldsymbol{\alpha}}_{k-1}}$ for $\rho \geq 0$. We also define $q(\boldsymbol{u}, \rho) = \boldsymbol{u}'H(\rho)\boldsymbol{u}$, where $\boldsymbol{u}$ is an $n$-dimensional vector. As shown in [18], $\partial F(\rho)^{-1}/\partial\rho = -F(\rho)^{-2}$ holds. Therefore, we have

$$\frac{\partial q(\boldsymbol{u}, \rho)}{\partial \rho} = -\boldsymbol{u}'G_{\widehat{\boldsymbol{\alpha}}_{k-1}}F(\rho)^{-2}G'_{\widehat{\boldsymbol{\alpha}}_{k-1}}\boldsymbol{u} \leq 0 \qquad \text{(A$\cdot$ 4)}$$

for any fixed $\boldsymbol{u}$. This implies that $q(\boldsymbol{u}, \rho)$ is a non-increasing function of $\rho$ for any fixed $\boldsymbol{u}$. Therefore, we have

$$q(\boldsymbol{u}, \rho) \leq q(\boldsymbol{u}, 0) \qquad \text{(A$\cdot$ 5)}$$

for any fixed $\boldsymbol{u}$ and $\rho \geq 0$. On the other hand, $H(0)$ is symmetric and idempotent for any $\widehat{\boldsymbol{\alpha}}_{k-1}$ since $\boldsymbol{g}_1, \ldots, \boldsymbol{g}_n$ are linearly independent. This implies that the rank of $H(0)$ is $k-1$, and thus $k-1$ of its eigenvalues are equal to 1 and the remaining $n-(k-1)$ eigenvalues are equal to 0; e.g., pp.50–51, [19]. We thus have $q(\boldsymbol{u},0)/\|\boldsymbol{u}\|^2 \le 1$ for any $\boldsymbol{u} \ne \boldsymbol{0}_n$ by p.533, [12].

By this fact and (A· 5), we have

$$\max_{\boldsymbol{u}\ne\boldsymbol{0}_n} q_{k-1}(\boldsymbol{u},\lambda)/\|\boldsymbol{u}\|^2 \le \max_{\boldsymbol{u}\ne\boldsymbol{0}_n} q(\boldsymbol{u},0)/\|\boldsymbol{u}\|^2 \le 1 \qquad \text{(A· 6)}$$

for a fixed $\lambda \ge 0$. Since $q_{k-1}(\boldsymbol{u},\lambda)/\|\boldsymbol{u}\|^2$ attains $\gamma_1$ for the eigenvector corresponding to $\gamma_1$, we have $\gamma_1 \le 1$. Here, it is easy to see that $\gamma_j < 1$ holds for some $j$.

We define $\boldsymbol{b}_l = Q'\boldsymbol{g}_l$. We then have

$$\boldsymbol{g}_l' H_{k-1}(\lambda)\boldsymbol{g}_l = \boldsymbol{b}_l'\Gamma\boldsymbol{b}_l < \|\boldsymbol{b}_l\|^2 = \|\boldsymbol{g}_l\|^2 \qquad \text{(A· 7)}$$

since $Q$ is an orthogonal matrix and $0 \le \gamma_j \le 1$ for any $j$ and $\gamma_j < 1$ for some $j$. We thus have

$$\boldsymbol{g}_l' P_{k-1}(\lambda)\boldsymbol{g}_l > 0 \qquad \text{(A· 8)}$$

by the definition of $P_{k-1}(\lambda)$. On the other hand, $\boldsymbol{g}_l' P_{k-1}(\lambda)^2\boldsymbol{g}_l \ge 0$ holds. $\boldsymbol{g}_l' P_{k-1}(\lambda)^2\boldsymbol{g}_l = 0$ holds if and only if $P_{k-1}(\lambda)\boldsymbol{g}_l = \boldsymbol{0}_n$ holds. The latter equation is equivalent to $H_{k-1}(\lambda)\boldsymbol{g}_l = \boldsymbol{g}_l$ by the definition of $P_{k-1}(\lambda)$. This contradicts (A· 7). Thus, we have

$$\boldsymbol{g}_l' P_{k-1}(\lambda)^2\boldsymbol{g}_l > 0. \qquad \text{(A· 9)}$$

By the above $Q$, we also have $Q'H_{k-1}(\lambda)^2 Q = \Gamma^2$ since $Q$ is an orthogonal matrix. We then have

$$\boldsymbol{g}_l' H_{k-1}(\lambda)^2\boldsymbol{g}_l = \boldsymbol{b}_l'\Gamma^2\boldsymbol{b}_l \le \boldsymbol{b}_l'\Gamma\boldsymbol{b}_l = \boldsymbol{g}_l' H_{k-1}(\lambda)\boldsymbol{g}_l \qquad \text{(A· 10)}$$

since $0 \le \gamma_j \le 1$. By the definition of $P_{\widehat{\alpha}_{k-1}}$, we have

$$\boldsymbol{g}_l' P_{\widehat{\alpha}_{k-1}}\boldsymbol{g}_l - \boldsymbol{g}_l' P^2_{\widehat{\alpha}_{k-1}}\boldsymbol{g}_l = \boldsymbol{g}_l' H_{k-1}(\lambda)\boldsymbol{g}_l - \boldsymbol{g}_l' H_{k-1}(\lambda)^2\boldsymbol{g}_l \ge 0. \qquad \text{(A· 11)}$$

## Appendix C: Properties of $\chi^2$ Random Variables

Let $Z_1,\ldots,Z_n$ be independent random variables from a common $N(0,1)$. Then, $Z_1^2,\ldots,Z_n^2$ are independent random variables from a common $\chi^2$ distribution with one degree of freedom. We define $f(t) = ((\sqrt{2}\Gamma(1/2))^{-1}t^{-1/2}\exp(-t/2)$ which is the probability density function of a $\chi^2$ distribution with one degree of freedom. We define $p(z) = \mathbb{P}\{Z_1 > z\}$. It is easy to see that

$$\lim_{z\to\infty}\frac{p(z)}{2f(z)} = 1 \qquad \text{(A· 12)}$$

by using $p(z) = \int_z^\infty f(t)dt$, $\int_0^\infty f(t) = 1$ and L'Hospital's rule. We define $\theta_n^2(\epsilon) = (2+\epsilon)\log n$, where $\epsilon$ is a constant that satisfies $\epsilon < 0$. We also define $p_n(\epsilon) = p(\theta_n^2(\epsilon))$ and $f_n(\epsilon) = f(\theta_n^2(\epsilon))$.

When $n$ is sufficiently large, by using (A· 12), we have

$$\begin{aligned}
1 - \mathbb{P}\left\{\max_{1\le i\le n} Z_i^2 > \theta_n^2(\epsilon)\right\} &= \mathbb{P}\left\{\max_{1\le i\le n} Z_i^2 < \theta_n^2(\epsilon)\right\}\\
&= \prod_{i=1}^{n}\mathbb{P}\left\{Z_i^2 < \theta_n^2(\epsilon)\right\}\\
&= \left(1 - \frac{np_n(\epsilon)}{n}\right)^n\\
&\le e^{-np_n(\epsilon)/(1-p_n(\epsilon))}\\
&\sim e^{-2nf_n(\epsilon)/(1-2f_n(\epsilon))}\\
&= \frac{\sqrt{n^\epsilon \log n}}{C_\epsilon}, \qquad \text{(A· 13)}
\end{aligned}$$

where $C_\epsilon$ is a positive constant and $\sim$ indicates the asymptotic equivalence.

**Katsuyuki Hagiwara** received the Ph.D. degree from Toyohashi University of Technology in 1995. The same year, he joined Mie University as a Research Associate in the Faculty of Engineering and is now an Associate Professor in the Faculty of Education. His research interests include machine learning, statistical model selection, and statistical signal processing.