IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol.E90-A, no.7, pp.1496–1499, 2007.
Updated on May 18, 2009.

Generalization Error Estimation for Non-Linear Learning Methods

Masashi Sugiyama (sugi@cs.titech.ac.jp)
Department of Computer Science, Tokyo Institute of Technology
2-12-1, O-okayama, Meguro-ku, Tokyo, 152-8552, Japan

Abstract

Estimating the generalization error is one of the key ingredients of supervised learning since a good generalization error estimator can be used for model selection. An unbiased generalization error estimator called the subspace information criterion (SIC) is shown to be useful for model selection, but its range of application is limited to linear learning methods. In this paper, we extend SIC to be applicable to non-linear learning.

Keywords

supervised learning, generalization capability, model selection, subspace information criterion, non-linear learning

1 Introduction

The goal of supervised learning is to estimate an unknown input-output relation from samples, which is mathematically formulated as a function approximation problem. If the learning target function is accurately learned, the output values for unlearned input points can be estimated. This is called the generalization capability. The level of generalization capability is evaluated by the 'closeness' between the learned function and the true function, i.e., the generalization error. We want to obtain the learned function that minimizes the generalization error. In order to obtain a better function, the model (e.g., type of basis functions, etc.) should be chosen appropriately, i.e., so that the generalization error is minimized.

However, since the true learning target function is unknown, the generalization error is not accessible. A standard approach to coping with this problem is to determine the model so that an estimator of the generalization error is minimized. The *subspace information criterion* (SIC) is one of the generalization error estimators for linear regression

[13]. SIC was shown to be a useful model selection criterion [14] and its theoretical properties from various aspects have been elucidated [11, 15]; in particular, SIC is shown to be a better estimator of the generalization error than standard estimators such as Akaike's information criterion or cross-validation in approximate linear regression [12]. However, the range of application of SIC was limited to linear learning methods—there are several useful learning methods that are non-linear, e.g., Huber's robust learning [7], sparse learning [18, 16, 2], or the support vector learning [17, 9]. In this paper, we extend SIC so that it can be used for estimating the generalization error of non-linear learning methods.

2 Problem Formulation

In this section, we formulate the linear regression problem.

Let us consider the regression problem of learning a real-valued function $f(\boldsymbol{x})$ defined on $\mathcal{D}(\subset \mathbb{R}^d)$ from training samples

$$\{(\boldsymbol{x}_i, y_i) \mid y_i = f(\boldsymbol{x}_i) + \epsilon_i\}_{i=1}^{n}, \tag{1}$$

where d is the dimension of the input vector $\boldsymbol{x}$, n is the number of training samples, and $\{\epsilon_i\}_{i=1}^{n}$ are i.i.d. noise with mean zero and variance σ^2. We employ the following linear regression model for learning:

$$\widehat{f}(\boldsymbol{x}) = \sum_{i=1}^{p} \alpha_i \varphi_i(\boldsymbol{x}), \tag{2}$$

where $\{\varphi_i(\boldsymbol{x})\}_{i=1}^{p}$ are fixed linearly independent functions, $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_p)^\top$ are parameters to be learned, and $^\top$ denotes the transpose of a vector/matrix. In practice, the above linear model may not be *correctly specified*, i.e., the learning target function $f(\boldsymbol{x})$ can not be expressed by the model (2). We define the generalization error of a learned function $\widehat{f}(\boldsymbol{x})$ by the expected squared error for test input points. We assume that the test input points are drawn independently from a distribution with density $q(\boldsymbol{x})$. Then the generalization error is expressed as

$$\int_{\mathcal{D}} \left(\widehat{f}(\boldsymbol{x}) - f(\boldsymbol{x})\right)^2 q(\boldsymbol{x}) d\boldsymbol{x}. \tag{3}$$

For making the following discussion simple, we subtract a constant C from the above quantity and define it as the generalization error G.

$$G = \int_{\mathcal{D}} \left(\widehat{f}(\boldsymbol{x}) - f(\boldsymbol{x})\right)^2 q(\boldsymbol{x}) d\boldsymbol{x} - C, \tag{4}$$

where

$$C = \int_{\mathcal{D}} f(\boldsymbol{x})^2 q(\boldsymbol{x}) d\boldsymbol{x}. \tag{5}$$

In the following theoretical discussions, we assume that $q(\boldsymbol{x})$ is known. Note that the generalization error G is still inaccessible even when $q(\boldsymbol{x})$ is known since the learning target function $f(\boldsymbol{x})$ is unknown. When $q(\boldsymbol{x})$ is unknown in practice, it may be estimated from unlabeled samples which are often abundantly available in some application domains.

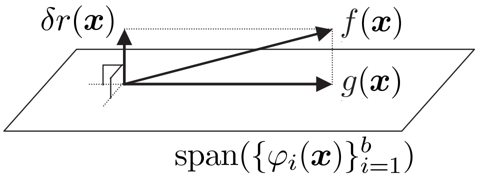
$\delta r(\boldsymbol{x})$
$f(\boldsymbol{x})$
$g(\boldsymbol{x})$
$\mathrm{span}(\{\varphi_i(\boldsymbol{x})\}_{i=1}^{b})$

Figure 1: Orthogonal decomposition of $f(\boldsymbol{x})$.

3 Subspace Information Criterion

Model selection is the problem of optimizing the model (e.g., the number and type of the basis functions $\{\varphi_i(\boldsymbol{x})\}_{i=1}^{p}$) so that the generalization error is minimized. In order to perform model selection, the inaccessible generalization error G has to be estimated. The *subspace information criterion* (SIC) [13] is one of the generalization error estimators. In this section, we derive SIC in a slightly generalized way.

Given that our linear regression model (2) is misspecified, the target function $f(\boldsymbol{x})$ is expressed as follows (see Figure 1):

$$f(\boldsymbol{x}) = g(\boldsymbol{x}) + \delta r(\boldsymbol{x}), \tag{6}$$

where $g(\boldsymbol{x})$ is the optimal approximation to $f(\boldsymbol{x})$ within the model (2):

$$g(\boldsymbol{x}) = \sum_{i=1}^{p} \alpha_i^* \varphi_i(\boldsymbol{x}). \tag{7}$$

$\boldsymbol{\alpha}^* = (\alpha_1^*, \alpha_2^*, \ldots, \alpha_p^*)^\top$ is the unknown optimal parameter vector under G:

$$\boldsymbol{\alpha}^* = \operatorname*{argmin}_{\boldsymbol{\alpha}} G. \tag{8}$$

$r(\boldsymbol{x})$ in Eq.(6) is the residual, which is orthogonal to $\{\varphi_i(\boldsymbol{x})\}_{i=1}^{p}$ under $q(\boldsymbol{x})$:

$$\int_{\mathcal{D}} r(\boldsymbol{x})\varphi_i(\boldsymbol{x})q(\boldsymbol{x})d\boldsymbol{x} = 0 \qquad \text{for } i = 1, 2, \ldots, p. \tag{9}$$

Without loss of generality, we assume that $r(\boldsymbol{x})$ is normalized in the following sense:

$$\int_{\mathcal{D}} r^2(\boldsymbol{x})q(\boldsymbol{x})d\boldsymbol{x} = 1. \tag{10}$$

Thus the function $r(\boldsymbol{x})$ governs the nature of the model error and δ is the possible magnitude of this error.

Given that $\widehat{f}(\boldsymbol{x})$ and $r(\boldsymbol{x})$ are orthogonal, G is written as

$$\begin{aligned} G &= \int_{\mathcal{D}} \widehat{f}(\boldsymbol{x})^2 q(\boldsymbol{x})d\boldsymbol{x} - 2\int_{\mathcal{D}} \widehat{f}(\boldsymbol{x})g(\boldsymbol{x})q(\boldsymbol{x})d\boldsymbol{x} \\ &= \|\widehat{\boldsymbol{\alpha}}\|_{\boldsymbol{U}}^2 - 2\langle \widehat{\boldsymbol{\alpha}}, \boldsymbol{\alpha}^* \rangle_{\boldsymbol{U}}, \end{aligned} \tag{11}$$

where $\|\boldsymbol{\alpha}\|_{\boldsymbol{U}}^2 = \boldsymbol{\alpha}^\top \boldsymbol{U}\boldsymbol{\alpha}$, $\langle \boldsymbol{\alpha}', \boldsymbol{\alpha}\rangle_{\boldsymbol{U}} = \boldsymbol{\alpha}^\top \boldsymbol{U}\boldsymbol{\alpha}'$, and $\boldsymbol{U}$ is the p-dimensional square matrix with the (i,j)-th element

$$U_{i,j} = \int_{\mathcal{D}} \varphi_i(\boldsymbol{x})\varphi_j(\boldsymbol{x})q(\boldsymbol{x})d\boldsymbol{x}. \tag{12}$$

A basic idea of SIC is to replace the unknown $\boldsymbol{\alpha}^*$ in Eq.(11) by the following linear estimator $\widehat{\boldsymbol{\alpha}}_u$:

$$\widehat{\boldsymbol{\alpha}}_u = \widehat{\boldsymbol{L}}_u \boldsymbol{y}, \tag{13}$$

where

$$\widehat{\boldsymbol{L}}_u = (\boldsymbol{X}^\top \boldsymbol{X})^{-1}\boldsymbol{X}^\top, \tag{14}$$

$$\boldsymbol{y} = (y_1, y_2, \ldots, y_n)^\top. \tag{15}$$

$\boldsymbol{X}$ is the $n \times p$ matrix with the (i,j)-th element

$$X_{i,j} = \varphi_j(\boldsymbol{x}_i). \tag{16}$$

However, simply replacing $\boldsymbol{\alpha}^*$ by $\widehat{\boldsymbol{\alpha}}_u$ causes a bias. Noting that $\boldsymbol{y}$ can be expressed as

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\alpha}^* + \delta\boldsymbol{r} + \boldsymbol{\epsilon}, \tag{17}$$

where

$$\boldsymbol{r} = (r(\boldsymbol{x}_1), r(\boldsymbol{x}_2), \ldots, r(\boldsymbol{x}_n))^\top, \tag{18}$$

the bias can be expressed as

$$\mathbb{E}_{\boldsymbol{\epsilon}}\left[\langle \widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\alpha}}_u\rangle_{\boldsymbol{U}} - \langle \widehat{\boldsymbol{\alpha}}, \boldsymbol{\alpha}^*\rangle_{\boldsymbol{U}}\right] = \mathbb{E}_{\boldsymbol{\epsilon}}\langle \widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{L}}_u\boldsymbol{\epsilon}\rangle_{\boldsymbol{U}} + \delta\mathbb{E}_{\boldsymbol{\epsilon}}\langle \widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{L}}_u\boldsymbol{r}\rangle_{\boldsymbol{U}}, \tag{19}$$

where $\mathbb{E}_{\boldsymbol{\epsilon}}$ denotes the expectation over the noise $\{\epsilon_i\}_{i=1}^n$ and

$$\boldsymbol{\epsilon} = (\epsilon_1, \epsilon_2, \ldots, \epsilon_n)^\top. \tag{20}$$

Based on this, we define 'preSIC' as follows.

$$\text{preSIC} = \|\widehat{\boldsymbol{\alpha}}\|_{\boldsymbol{U}}^2 - 2\langle \widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{L}}_u\boldsymbol{y}\rangle_{\boldsymbol{U}} + 2\mathbb{E}_{\boldsymbol{\epsilon}}\langle \widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{L}}_u\boldsymbol{\epsilon}\rangle_{\boldsymbol{U}}, \tag{21}$$

i.e., $\delta\mathbb{E}_{\boldsymbol{\epsilon}}\langle \widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{L}}_u\boldsymbol{r}\rangle_{\boldsymbol{U}}$ is ignored. If the third term $\mathbb{E}_{\boldsymbol{\epsilon}}\langle \widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{L}}_u\boldsymbol{\epsilon}\rangle_{\boldsymbol{U}}$ can be computed (or approximated) from the training samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, we can use preSIC for model selection.

Let us focus on *linear learning*, i.e., the learned parameter vector $\widehat{\boldsymbol{\alpha}}$ is given by

$$\widehat{\boldsymbol{\alpha}} = \boldsymbol{L}\boldsymbol{y}, \tag{22}$$

where $\boldsymbol{L}$ is a $p \times n$ matrix which is independent of the noise $\{\epsilon_i\}_{i=1}^n$. This includes popular *ℓ_2-norm regularization learning* [6, 5]:

$$\min_{\boldsymbol{\alpha}} \left[\sum_{i=1}^n \left(\widehat{f}(\boldsymbol{x}_i) - y_i\right)^2 + \lambda\|\boldsymbol{\alpha}\|^2\right], \tag{23}$$

where λ (≥ 0) is a tuning parameter. The learned parameter vector $\widehat{\boldsymbol{\alpha}}$ is given by Eq.(22) with

$$\boldsymbol{L} = (\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I})^{-1}\boldsymbol{X}^\top, \tag{24}$$

where $\boldsymbol{I}$ is the identity matrix.

Proposition 1 ([13]) *For linear learning* (22), *we have*

$$\mathbb{E}_{\boldsymbol{\epsilon}}\langle\widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{L}}_u\boldsymbol{\epsilon}\rangle_{\boldsymbol{U}} = \sigma^2\mathrm{tr}(\boldsymbol{U}\boldsymbol{L}\widehat{\boldsymbol{L}}_u^{\top}). \tag{25}$$

Based on this proposition, SIC for a linear learning method is given by

$$\mathrm{SIC} = \|\widehat{\boldsymbol{\alpha}}\|_{\boldsymbol{U}}^2 - 2\langle\widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{L}}_u\boldsymbol{y}\rangle_{\boldsymbol{U}} + 2\sigma^2\mathrm{tr}(\boldsymbol{U}\boldsymbol{L}\widehat{\boldsymbol{L}}_u^{\top}). \tag{26}$$

SIC is shown to satisfy[1]

$$\mathbb{E}_{\boldsymbol{\epsilon}}\mathrm{SIC} = \mathbb{E}_{\boldsymbol{\epsilon}}G + \mathcal{O}_p(\delta n^{-\frac{1}{2}}), \tag{27}$$

where $\mathcal{O}_p$ denotes the asymptotic order in probability [12]. This means that, SIC is an exact unbiased estimator of the expected generalization error if the model is correctly specified (i.e., the model error δ is zero); otherwise, it is an asymptotic unbiased estimator in general, where the bias is proportional to the model error δ.

The goal of this paper is to extend SIC so that it can be used for estimating the generalization error of non-linear learning methods.

4 Extension to Non-Linear Learning

In this section, we extend the range of application of SIC to non-linear learning methods.

4.1 Affine Learning

We start from a simple non-linear learning method called *affine learning*, i.e., for a $p \times n$ matrix $\boldsymbol{L}$ and a p-dimensional vector $\boldsymbol{c}$, both of which are independent of the noise $\{\epsilon_i\}_{i=1}^n$, the learned parameter vector $\widehat{\boldsymbol{\alpha}}$ is given by

$$\widehat{\boldsymbol{\alpha}} = \boldsymbol{L}\boldsymbol{y} + \boldsymbol{c}. \tag{28}$$

This includes *additive regularization learning* [8]:

$$\min_{\boldsymbol{\alpha}} \left[\sum_{i=1}^n \left(\widehat{f}(\boldsymbol{x}_i) - y_i - \lambda_i\right)^2 + \|\boldsymbol{\alpha}\|^2\right], \tag{29}$$

where $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_n)^{\top}$ is tuning parameters. The learned parameter vector $\widehat{\boldsymbol{\alpha}}$ is given by Eq.(28) with

$$\boldsymbol{L} = (\boldsymbol{X}^{\top}\boldsymbol{X} + \boldsymbol{I})^{-1}\boldsymbol{X}^{\top}, \tag{30}$$

$$\boldsymbol{c} = (\boldsymbol{X}^{\top}\boldsymbol{X} + \boldsymbol{I})^{-1}\boldsymbol{X}^{\top}\boldsymbol{\lambda}. \tag{31}$$

[1] Under some kernel regression scenarios, SIC becomes exactly unbiased with finite samples irrespective of the model error δ [11]. Since this paper includes the kernel regression setting of the reference [11] as a special case, we can enjoy this excellent property.

Lemma 1 *For affine learning* (28)*, we have*

$$\mathbb{E}_{\boldsymbol{\epsilon}}\langle \widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{L}}_u \boldsymbol{\epsilon}\rangle_{\boldsymbol{U}} = \sigma^2 \mathrm{tr}(\boldsymbol{U}\boldsymbol{L}\widehat{\boldsymbol{L}}_u^\top). \tag{32}$$

Proof: Given that $\mathbb{E}_{\boldsymbol{\epsilon}}\boldsymbol{\epsilon} = \boldsymbol{0}$ and $\mathbb{E}_{\boldsymbol{\epsilon}}[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top] = \sigma^2\boldsymbol{I}$, we have Eq.(32). ■

This lemma implies that we can still use the same SIC (26) for affine learning without any performance degradation.

4.2 Smooth Non-Linear Learning

Let us consider a *smooth non-linear learning* method, i.e., using an *almost differentiable* [10] operator $\boldsymbol{L}$, $\widehat{\boldsymbol{\alpha}}$ is given by

$$\widehat{\boldsymbol{\alpha}} = \boldsymbol{L}(\boldsymbol{y}). \tag{33}$$

This includes *Huber's robust estimation* [7]:

$$\min_{\boldsymbol{\alpha}} \left[\sum_{i=1}^{n} \rho_\tau(\widehat{f}(\boldsymbol{x}_i) - y_i)^2 \right], \tag{34}$$

where τ (> 0) is a tuning parameter and

$$\rho_\tau(y) = \begin{cases} y^2/2 & (|y| \leq \tau), \\ \tau|y| - y^2/2 & (|y| > \tau). \end{cases} \tag{35}$$

Note that $\rho_\tau(y)$ is twice almost differentiable, which yields a once almost differentiable operator $\boldsymbol{L}$.

Lemma 2 *Let $\boldsymbol{H}$ be the n-dimensional square matrix with the (i,j)-th element*

$$\nabla_i[\widehat{\boldsymbol{L}}_u^\top \boldsymbol{U}\boldsymbol{L}]_j(\boldsymbol{y}), \tag{36}$$

where ∇_i is the partial derivative operator with respect to the i-th element and $[\widehat{\boldsymbol{L}}_u^\top \boldsymbol{U}\boldsymbol{L}]_j(\boldsymbol{y})$ denotes the j-th element of the vector-valued function $[\widehat{\boldsymbol{L}}_u^\top \boldsymbol{U}\boldsymbol{L}](\boldsymbol{y})$. Suppose the noise $\{\epsilon_i\}_{i=1}^n$ is Gaussian. Then, for smooth non-linear learning (33)*, we have*

$$\mathbb{E}_{\boldsymbol{\epsilon}}\langle \widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{L}}_u \boldsymbol{\epsilon}\rangle_{\boldsymbol{U}} = \sigma^2 \mathbb{E}_{\boldsymbol{\epsilon}} \mathrm{tr}(\boldsymbol{H}). \tag{37}$$

Proof: For an n-dimensional centered i.i.d. Gaussian vector $\boldsymbol{\epsilon}$ and for any almost differentiable function $h(\boldsymbol{\epsilon})$ defined on $\mathbb{R}^n$, the following *Stein's identity* holds [10]:

$$\mathbb{E}_{\boldsymbol{\epsilon}}[\epsilon_i h(\boldsymbol{\epsilon})] = \sigma^2 \mathbb{E}_{\boldsymbol{\epsilon}}[\nabla_i h(\boldsymbol{\epsilon})]. \tag{38}$$

Let $\boldsymbol{h}(\boldsymbol{\epsilon}) = [\widehat{\boldsymbol{L}}_u^\top \boldsymbol{U}\boldsymbol{L}](\boldsymbol{y})$. Since $\boldsymbol{L}(\boldsymbol{y})$ is almost differentiable, $\boldsymbol{h}(\boldsymbol{\epsilon})$ is also almost differentiable. Then an element-wise application of Eq.(38) to a vector-valued function $\boldsymbol{h}(\boldsymbol{\epsilon})$ establishes Eq.(37). ■

Based on this lemma, we define SIC for smooth non-linear learning as

$$\mathrm{SIC} = \|\widehat{\boldsymbol{\alpha}}\|_{\boldsymbol{U}}^2 - 2\langle \widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{L}}_u \boldsymbol{y} \rangle_{\boldsymbol{U}} + 2\sigma^2 \mathrm{tr}(\boldsymbol{H}), \tag{39}$$

which still maintains the same unbiasedness property (27). It is easy to confirm that Eq.(39) is reduced to the original SIC (26) when $\widehat{\boldsymbol{\alpha}}$ is obtained by linear learning (22). Therefore, the above SIC may be regarded as a natural extension of the original one.

4.3 General Non-Linear Learning

Finally, let us consider general non-linear learning methods, i.e., for a general non-linear operator $\boldsymbol{L}$, the learned parameter vector $\widehat{\boldsymbol{\alpha}}$ is given by

$$\widehat{\boldsymbol{\alpha}} = L(\boldsymbol{y}). \tag{40}$$

This includes ℓ_1*-norm regularization learning* [18, 16, 2]:

$$\min_{\boldsymbol{\alpha}} \left[\sum_{i=1}^{n} \left(\widehat{f}(\boldsymbol{x}_i) - y_i \right)^2 + \lambda \|\boldsymbol{\alpha}\|_1 \right], \tag{41}$$

where

$$\|\boldsymbol{\alpha}\|_1 = \sum_{i=1}^{p} |\alpha_i|. \tag{42}$$

For general non-linear learning, we estimate the third term $\mathbb{E}_{\boldsymbol{\epsilon}} \langle \widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{L}}_u \boldsymbol{\epsilon} \rangle_{\boldsymbol{U}}$ in preSIC (21) using the *bootstrap* method [3, 4]:

$$\mathbb{E}_{\boldsymbol{\epsilon}} \langle \widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{L}}_u \boldsymbol{\epsilon} \rangle_{\boldsymbol{U}} \approx \mathbb{E}_{\boldsymbol{\epsilon}}^b \langle \widehat{\boldsymbol{\alpha}}^b, \widehat{\boldsymbol{L}}_u \widehat{\boldsymbol{\epsilon}}^b \rangle_{\boldsymbol{U}}, \tag{43}$$

where $\mathbb{E}_{\boldsymbol{\epsilon}}^b$ denotes the expectation over the bootstrap replication and $\widehat{\boldsymbol{\alpha}}^b$ and $\widehat{\boldsymbol{\epsilon}}^b$ are the learned parameter vector and the noise vector estimated from the bootstrap samples, respectively. More specifically, we compute $\mathbb{E}_{\boldsymbol{\epsilon}}^b \langle \widehat{\boldsymbol{\alpha}}^b, \widehat{\boldsymbol{L}}_u \widehat{\boldsymbol{\epsilon}}^b \rangle_{\boldsymbol{U}}$ by *bootstrapping residuals* as follows.

1. Obtain the learned parameter vector $\widehat{\boldsymbol{\alpha}}$ using the training samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ as usual.
2. Estimate the noise by $\{\widehat{\epsilon}_i \mid \widehat{\epsilon}_i = y_i - \widehat{f}(\boldsymbol{x}_i)\}_{i=1}^n$.
3. Create bootstrap noise samples $\{\widehat{\epsilon}_i^b\}_{i=1}^n$ by sampling with replacement from $\{\widehat{\epsilon}_i\}_{i=1}^n$.
4. Obtain the learned parameter vector $\widehat{\boldsymbol{\alpha}}^b$ using the bootstrap samples $\{(\boldsymbol{x}_i, y_i^b) \mid y_i^b = \widehat{f}(\boldsymbol{x}_i) + \widehat{\epsilon}_i^b\}_{i=1}^n$.
5. Calculate $\langle \widehat{\boldsymbol{\alpha}}^b, \widehat{\boldsymbol{L}}_u \widehat{\boldsymbol{\epsilon}}^b \rangle_{\boldsymbol{U}}$.
6. Repeat 3. to 5. for a number of times and output the mean of $\langle \widehat{\boldsymbol{\alpha}}^b, \widehat{\boldsymbol{L}}_u \widehat{\boldsymbol{\epsilon}}^b \rangle_{\boldsymbol{U}}$.

Based on this procedure, *bootstrap-approximated SIC* (BASIC) is defined as

$$\mathrm{BASIC} = \|\widehat{\boldsymbol{\alpha}}\|_{\boldsymbol{U}}^2 - 2\langle \widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{L}}_u \boldsymbol{y} \rangle_{\boldsymbol{U}} + 2\mathbb{E}_{\boldsymbol{\epsilon}}{}^b \langle \widehat{\boldsymbol{\alpha}}^b, \widehat{\boldsymbol{L}}_u \widehat{\boldsymbol{\epsilon}}^b \rangle_{\boldsymbol{U}}, \tag{44}$$

which may give an approximately unbiased estimate of the expected generalization error.

5 Conclusions and Future Prospects

We extended the range of application of SIC so that the generalization error of various useful non-linear learning methods can be estimated. Currently, the range of application of Lemma 2 is restricted to Gaussian noise. An important future direction is to alleviate the condition using a generalized Stein's identity (e.g., [1]). Also, theoretical analysis of the approximation quality of BASIC is still open currently.

Acknowledgments

The author would like to thank Nguyen Huu Bach and Ryo Nitta for their fruitful comments. This work is partially supported by MEXT (Grant-in-Aid for Young Scientists 17700142, Grant-in-Aid for Scientific Research (B) 18300057, Scholarship 053155).

References

[1] G. Blanchard, M. Kawanabe, M. Sugiyama, V. Spokoiny, and K.R. Müller, "In search of non-Gaussian components of a high-dimensional distribution," Journal of Machine Learning Research, vol.7, pp.247–282, Feb. 2006.

[2] S.S. Chen, D.L. Donoho, and M.A. Saunders, "Atomic decomposition by basis pursuit," SIAM Journal on Scientific Computing, vol.20, no.1, pp.33–61, 1998.

[3] B. Efron, "Bootstrap methods: Another look at the jackknife," The Annals of Statistics, vol.7, no.1, pp.1–26, 1979.

[4] B. Efron and R.J. Tibshirani, An Introduction to the Bootstrap, Chapman & Hall, New York, 1993.

[5] T. Evgeniou, M. Pontil, and T. Poggio, "Regularization networks and support vector machines," Advances in Computational Mathematics, vol.13, no.1, pp.1–50, 2000.

[6] F. Girosi, M. Jones, and T. Poggio, "Regularization theory and neural networks architectures," Neural Computation, vol.7, no.2, pp.219–269, 1995.

[7] P.J. Huber, Robust Statistics, Wiley, New York, 1981.

[8] K. Pelckmans, J.A.K. Suykens, and B. De Moor, "Additive regularization: Fusion of training and validation levels in kernel methods," Machine Learning, vol.62, no.3, pp.217–252, 2004.

[9] B. Schölkopf and A.J. Smola, Learning with Kernels, MIT Press, Cambridge, MA, 2002.

[10] C.M. Stein, “Estimation of the mean of a multivariate normal distribution,” The Annals of Statistics, vol.9, no.6, pp.1135–1151, 1981.

[11] M. Sugiyama and K.R. Müller, “The subspace information criterion for infinite dimensional hypothesis spaces,” Journal of Machine Learning Research, vol.3, pp.323–359, Nov. 2002.

[12] M. Sugiyama and K.R. Müller, “Input-dependent estimation of generalization error under covariate shift,” Statistics & Decisions, vol.23, no.4, pp.249–279, 2005.

[13] M. Sugiyama and H. Ogawa, “Subspace information criterion for model selection,” Neural Computation, vol.13, no.8, pp.1863–1889, 2001.

[14] M. Sugiyama and H. Ogawa, “Theoretical and experimental evaluation of the subspace information criterion,” Machine Learning, vol.48, no.1/2/3, pp.25–50, 2002.

[15] M. Sugiyama, Y. Okabe, and H. Ogawa, “Perturbation analysis of a generalization error estimator,” Neural Information Processing - Letters and Reviews, vol.2, no.2, pp.33–38, 2004.

[16] R. Tibshirani, “Regression shrinkage and selection via the lasso,” Journal of the Royal Statistical Society, Series B, vol.58, no.1, pp.267–288, 1996.

[17] V.N. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.

[18] P.M. Williams, “Bayesian regularization and pruning using a Laplace prior,” Neural Computation, vol.7, no.1, pp.117–143, 1995.