Subspace Information Criterion for Non-Quadratic Regularizers — Model Selection for Sparse Regressors

Koji Tsuda[*+], Masashi Sugiyama[†] and Klaus-Robert Müller[*‡]

[*]GMD FIRST, Kekuléstr. 7, 12489 Berlin, Germany

[+] AIST Computational Biology Research Center,
2-41-6, Aomi, Koto-ku, Tokyo, 135-0064, Japan

[†]Tokyo Institute of Technology,
2-12-1, O-okayama, Meguro-ku, Tokyo, 152-8552, Japan

[‡]University of Potsdam, Am Neuen Palais 10, 14469 Potsdam, Germany

koji.tsuda@aist.go.jp, sugi@og.cs.titech.ac.jp, klaus@first.gmd.de

Abstract

Non-quadratic regularizers, in particular the ℓ_1 norm regularizer can yield sparse solutions that generalize well. In this work we propose the Generalized Subspace Information Criterion (GSIC) that allows to predict the generalization error for this useful family of regularizers. We show that under some technical assumptions GSIC is an asymptotically unbiased estimator of the generalization error. GSIC is demonstrated to have a good performance in experiments with the ℓ_1 norm regularizer as we compare with the Network Information Criterion and cross-validation in relatively large sample cases. However in the small sample case, GSIC tends to fail to capture the optimal model due to its large variance. Therefore, also a biased version of GSIC is introduced, which achieves reliable model selection in the relevant and challenging scenario of high dimensional data and few samples.

1 Introduction

Supervised learning techniques allow to estimate underlying unknown statistical input-output relations from given training data [1, 2, 3]. In this process one has to be careful not to overfit the training data, but to estimate the underlying statistical data generation process, such that the learning machine generalizes well, i.e. that it gives a good estimate even for unseen data.

One way to avoid overfitting is to restrict the function class from which the estimators are chosen. Thus, one introduces a preference from complicated models towards simpler models for example by choosing a model with a small VC dimension [1] or by introducing regularization [4]. Intuitively this amounts to selecting a smoother model.

In this paper we will consider regularization for enhancing the generalization capability. Here the parameters $\boldsymbol{\theta}$ of the learning machine are determined such that a weighted sum of the training error and the regularization term R

$$\text{Error}(\boldsymbol{\theta}) = \text{TrainingError}(\boldsymbol{\theta}) + \lambda R(\boldsymbol{\theta}) \tag{1}$$

is minimized, where λ is called the *regularization constant*. If the regularization constant λ is too large, then the estimator is *under-fitting*, the estimation is too smooth and the generalization error becomes large. If λ is selected too small, then overfitting and high frequency ("wiggly") estimators result. Therefore, the problem of model selection, i.e. in our case determining the value of the regularization constant is *essential* for good generalization performance. There is a large body of literature of how to choose the regularization constant (e.g. for neural networks see [2, 5, 6]). The ideal criterion would be the generalization error itself, or approximations thereof, e.g. in a worst or average case setting. The former considers the worst generalization error achieved on all possible training sets (see e.g. methods based on VC theory [7, 1]). The latter considers ensemble averages over all possible training sets, for example the Network Information Criterion (NIC) [8, 9] or the Subspace Information Criterion (SIC) [10]. Furthermore there are very successful criteria such as cross validation [11], C_L [12] or the Bayesian evidence framework [13, 14], which approximately evaluate the ensemble error using the training data. In this paper, we will focus on prediction methods for the ensemble average of the generalization error.

The prediction of the generalization error becomes easier if *additional unlabeled* input data points are known. NIC – a generalization of Akaike's information criterion [15] – is a typical method which does *not* make use of the distribution of unlabeled additional data points. It only assumes that all data has essentially the same distribution as the training samples. For example in text classification [16] many additional unlabeled samples are available, so an accurate estimation of the input distribution beyond the training data is possible. SIC – a generalization of C_L – makes use of additional unlabeled data and therefore has been shown to perform better than NIC, particularly in the small sample setting [10]. The technical feature of SIC is that it predicts the generalization error by utilizing a *reference estimator*, which is an unbiased estimate of the true parameter. SIC

was so far only applicable to linear regression with *quadratic* regularizers, which includes e.g. *weight decay* (see [2, 17, 18]).

Recently, sparsity inducing non-quadratic regularizers have become rather popular since with still good generalization properties [19, 20, 21, 22, 23, 24, 25] sparse solutions (i.e. most of the model parameters become zero) are found in the training process.[1] Often they are based on l_1 regularization. Since such regularization terms are non-quadratic, the original SIC criterion cannot be applied to them.

In this work we therefore propose the Generalized Subspace Information Criterion (GSIC) that allows to predict the generalization error for the family of non-quadratic regularizers. Among several other interesting theoretical properties, we will – under several assumptions – show that GSIC is an asymptotically *unbiased* estimator of the generalization error.

In experiments with relatively large samples, GSIC achieves a good performance as we compare with NIC and cross-validation. However, in small sample cases, GSIC tends to fail to capture the optimal model due to its large variance. To alleviate this problem, we introduce a biased version of GSIC, which is derived from a reference estimator regularized by a quadratic regularizer. This biased version (GSICb) introduces yet another model selection problem: determining the regularization constant of the reference estimator. But, since a quadratic regularizer is used here, the regularization constant can be determined by efficient algorithms [5]. In experiments using an ℓ_1 norm regularizer, GSICb shows an excellent performance, when compared to NIC and cross-validation.

The rest of this paper is organized as follows: In Sec. 2, we formulate the problem of generalization error prediction in detail. In Sec. 3, the generalized SIC for non-quadratic regularizer is proposed, and its asymptotic bias is investigated. Sec. 4 introduces a biased version of GSIC for small sample cases. Sec. 5 considers the application of GSIC to sparse regressors. Experiments in Sec. 6 give a comparison of our method with NIC and cross validation. Finally, Sec. 7 gives concluding remarks.

2 Preliminaries

In a linear regression problem, a target function is approximated by a parametric model which is linear in parameters. Let us assume that the target function $f(\boldsymbol{x})$, $\boldsymbol{x} \in \mathbb{R}^d$, is contained in a parametric model

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{i=1}^{p} \theta_i \phi_i(\boldsymbol{x}), \tag{2}$$

[1] In principle one could select the parameters afterwards by feature selection techniques [26]. However, such a two-stage scheme has often a more complex algorithmic structure and is harder to analyze. The ℓ_1 regularizer appears more simple.

where $\phi_i : \mathbb{R}^d \to \mathbb{R}$ is a given (nonlinear) basis function and $\boldsymbol{\theta} \in \mathbb{R}^p$ is the parameter vector. Then, we can describe $f(\boldsymbol{x})$ as

$$f(\boldsymbol{x}) = \sum_{i=1}^{p} \theta_i^* \phi_i(\boldsymbol{x}), \tag{3}$$

where θ_i^* is the true parameter.[2] The training examples consist of input points $\boldsymbol{x}_i \in \mathbb{R}^d$ and the corresponding output $y_i \in \mathbb{R}$, which are degraded by additive noise ϵ_i:

$$y_i = f(\boldsymbol{x}_i) + \epsilon_i. \tag{4}$$

We assume that all random variables $\{\epsilon_i\}_{i=1}^n$ are independent and subject to the same symmetric distribution with mean zero and variance σ^2. In this paper, we focus on the case where the parameter $\boldsymbol{\theta}$ is determined by finding $\boldsymbol{\theta}$ that minimizes a weighted sum of squared errors and a (twice differentiable) *regularization term* $R(\boldsymbol{\theta})$

$$L_r = \frac{1}{n} \sum_{i=1}^{n} (f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) - y_i)^2 + \lambda R(\boldsymbol{\theta}), \tag{5}$$

where λ is called the *regularization constant*. Let us define $\hat{\boldsymbol{\theta}}$ as the solution of the optimization problem:

$$\hat{\boldsymbol{\theta}} = \operatorname*{argmin}_{\boldsymbol{\theta}} L_r(\boldsymbol{\theta}). \tag{6}$$

The generalization error of $\hat{\boldsymbol{\theta}}$ is

$$E_{\boldsymbol{x}}[(f(\boldsymbol{x}) - f_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}))^2] = \int (f(\boldsymbol{x}) - f_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}))^2 q(\boldsymbol{x}) d\boldsymbol{x}, \tag{7}$$

where $q(\boldsymbol{x})$ denotes the distribution of input $\boldsymbol{x}$. Let us assume that the solution of (5) is unique. Then, the solution $\hat{\boldsymbol{\theta}}$ is considered an implicit function of training examples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$:

$$\hat{\boldsymbol{\theta}}(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n, y_1, \cdots, y_n). \tag{8}$$

In model selection, the optimal λ should be determined so that the generalization error is minimized. However, since $\hat{\boldsymbol{\theta}}$ depends on random variables y_i, the generalization error (7) is also a random variable. In order to compare two random variables, we focus on the mean only. The mean generalization error

$$J_G = E_{\boldsymbol{\epsilon}} E_{\boldsymbol{x}}[(f(\boldsymbol{x}) - f_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}))^2], \tag{9}$$

[2] Assuming there is a true parameter may seem restrictive, but this is actually not a strong condition, because in sparse regression it is common to start with a large number of basis functions and to subsequently reduce the number of them by minimizing a sparseness inducing error function.

is called *ensemble average*, where $\hat{\boldsymbol{\theta}}(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n, y_1, \cdots, y_n)$ is abbreviated as $\hat{\boldsymbol{\theta}}$ and $E_{\boldsymbol{\epsilon}} := E_{\epsilon_1} \cdots E_{\epsilon_n}$.

For the sake of a better geometrical understanding, we define the inner product in parameter space as

$$\begin{aligned} \langle \boldsymbol{\theta}, \boldsymbol{\theta}' \rangle &= E_{\boldsymbol{x}}[(\sum_{j=1}^{p} \theta_j \phi_j(\boldsymbol{x}))(\sum_{k=1}^{p} \theta'_k \phi_k(\boldsymbol{x}))] \\ &= \boldsymbol{\theta}^T P \boldsymbol{\theta}', \end{aligned} \tag{10}$$

where P is the matrix whose (i, j) element is given as

$$P_{ij} = E_{\boldsymbol{x}}[\phi_i(\boldsymbol{x})\phi_j(\boldsymbol{x})]. \tag{11}$$

Then we can rewrite the ensemble average of the generalization error using the norm of parameter space[3] as

$$J_G = E_{\boldsymbol{\epsilon}} \|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*\|^2. \tag{12}$$

The matrix P can be exactly calculated if we know the input distribution $q(\boldsymbol{x})$. If $q(\boldsymbol{x})$ is unknown, P can be estimated, e.g. by using the unlabeled samples $\{\boldsymbol{x}'_k\}_{k=1}^m$ [16] or one can assume that $q(\boldsymbol{x})$ is the uniform distribution over some domain.

3 Generalization Error Prediction

In this section, we derive a generalization error prediction method called Generalized Subspace Information Criterion (GSIC).

3.1 Basic Idea

Fig. 1 illustrates the idea. With respect to different training sets, the parameter $\hat{\boldsymbol{\theta}}$ takes various values and forms a distribution, where $\boldsymbol{\theta}^m$ is the mean of $\hat{\boldsymbol{\theta}}$, i.e., $\boldsymbol{\theta}^m = E_{\boldsymbol{\epsilon}}[\hat{\boldsymbol{\theta}}]$. The generalization error J_G is the average distance between $\hat{\boldsymbol{\theta}}$ and the underlying true solution $\boldsymbol{\theta}^*$. Because there is no information about $\boldsymbol{\theta}^*$, we introduce another parameter $\hat{\boldsymbol{\theta}^u}$ such that $\hat{\boldsymbol{\theta}^u}$ is an unbiased estimate of $\boldsymbol{\theta}^*$:

$$E_{\boldsymbol{\epsilon}}[\hat{\boldsymbol{\theta}^u}] = \boldsymbol{\theta}^*. \tag{13}$$

A typical choice of $\hat{\boldsymbol{\theta}^u}$ is the least mean squares estimator (i.e. without the regularizer) [10].[4] Then the distance between $\hat{\boldsymbol{\theta}}$ and $\hat{\boldsymbol{\theta}^u}$ (the broken line in Fig. 1) gives a rough estimate of the generalization error. We will derive an unbiased estimator of the generalization error by adding modification terms to this distance. Note that this technique to use an unbiased estimator was first introduced in SIC [10].

[3] This is actually a seminorm, because the norm $\|\boldsymbol{\theta}\|$ will vanish for nonzero $\boldsymbol{\theta}$ if $\boldsymbol{\theta}$ lies in the null space of P. However, our theoretical discussions hold even when P has a null space.

[4] Notice that the least mean squares estimator is unbiased only when $n \geq p$.

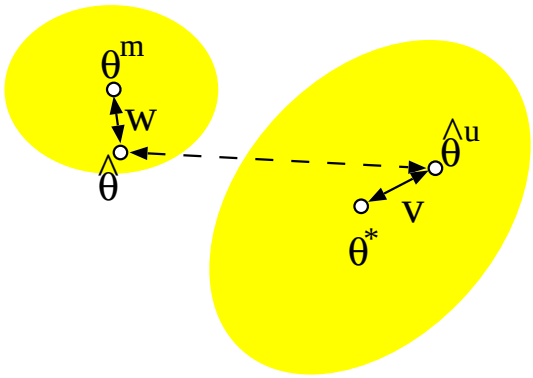


Figure 1: Basic idea for evaluating the generalization error.

3.2 Generalized Subspace Information Criterion

In this section, we will derive an unbiased estimator of J_G. J_G can be decomposed into the *bias* and *variance* (see also [27, 28]) as

$$\begin{aligned} E_\epsilon\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*\|^2 &= E_\epsilon\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^m + \boldsymbol{\theta}^m - \boldsymbol{\theta}^*\|^2 \\ &= E_\epsilon\|\boldsymbol{w} + \boldsymbol{\theta}^m - \boldsymbol{\theta}^*\|^2 \\ &= \|\boldsymbol{\theta}^m - \boldsymbol{\theta}^*\|^2 + 2E_\epsilon\langle\boldsymbol{\theta}^m - \boldsymbol{\theta}^*, \boldsymbol{w}\rangle + E_\epsilon\|\boldsymbol{w}\|^2 \\ &= \|\boldsymbol{\theta}^m - \boldsymbol{\theta}^*\|^2 + E_\epsilon\langle\boldsymbol{w}, \boldsymbol{w}\rangle, \end{aligned} \tag{14}$$

where $\boldsymbol{w} := \hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^m$. The bias term can be expressed by using $\|\hat{\boldsymbol{\theta}} - \hat{\boldsymbol{\theta}^u}\|^2$ as

$$\begin{aligned} \|\boldsymbol{\theta}^m - \boldsymbol{\theta}^*\|^2 &= \|\hat{\boldsymbol{\theta}} - \hat{\boldsymbol{\theta}^u}\|^2 - \|\hat{\boldsymbol{\theta}} - \hat{\boldsymbol{\theta}^u}\|^2 + \|\boldsymbol{\theta}^m - \boldsymbol{\theta}^*\|^2 \\ &= \|\hat{\boldsymbol{\theta}} - \hat{\boldsymbol{\theta}^u}\|^2 - \|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^m + \boldsymbol{\theta}^m - \hat{\boldsymbol{\theta}^u} + \boldsymbol{\theta}^* - \boldsymbol{\theta}^*\|^2 + \|\boldsymbol{\theta}^m - \boldsymbol{\theta}^*\|^2 \\ &= \|\hat{\boldsymbol{\theta}} - \hat{\boldsymbol{\theta}^u}\|^2 - \|\boldsymbol{w} + \boldsymbol{\theta}^m - \boldsymbol{v} - \boldsymbol{\theta}^*\|^2 + \|\boldsymbol{\theta}^m - \boldsymbol{\theta}^*\|^2 \\ &= \|\hat{\boldsymbol{\theta}} - \hat{\boldsymbol{\theta}^u}\|^2 - \|\boldsymbol{w} - \boldsymbol{v}\|^2 - 2\langle\boldsymbol{w} - \boldsymbol{v}, \boldsymbol{\theta}^m - \boldsymbol{\theta}^*\rangle, \end{aligned} \tag{15}$$

where $\boldsymbol{v} := \hat{\boldsymbol{\theta}^u} - \boldsymbol{\theta}^*$. The second and third terms in (15) can not be directly evaluated, so we average out these terms. Then the second term yields

$$-E_\epsilon\|\boldsymbol{w} - \boldsymbol{v}\|^2 = -E_\epsilon\langle\boldsymbol{w}, \boldsymbol{w}\rangle + 2E_\epsilon\langle\boldsymbol{w}, \boldsymbol{v}\rangle - E_\epsilon\langle\boldsymbol{v}, \boldsymbol{v}\rangle, \tag{16}$$

and the third term vanishes as

$$E_\epsilon\langle\boldsymbol{w} - \boldsymbol{v}, \boldsymbol{\theta}^m - \boldsymbol{\theta}^*\rangle = \langle E_\epsilon\boldsymbol{w} - E_\epsilon\boldsymbol{v}, \boldsymbol{\theta}^m - \boldsymbol{\theta}^*\rangle = 0, \tag{17}$$

because $E_\epsilon\boldsymbol{w} = E_\epsilon\boldsymbol{v} = \boldsymbol{0}$. This approximation yields the unbiased estimator of J_G called the *Generalized Subspace Information Criterion*.

Definition 1 (Generalized Subspace Information Criterion) *The following functional is defined as the Generalized Subspace Information Criterion:*

$$GSIC = \|\hat{\boldsymbol{\theta}} - \hat{\boldsymbol{\theta}^u}\|^2 + 2E_{\boldsymbol{\epsilon}}\langle \boldsymbol{w}, \boldsymbol{v}\rangle - E_{\boldsymbol{\epsilon}}\langle \boldsymbol{v}, \boldsymbol{v}\rangle. \tag{18}$$

Note that the proposed GSIC includes SIC as a special case.

3.3 GSIC for Quadratic Regularizers

For calculating (18), an unbiased estimate $\hat{\boldsymbol{\theta}^u}$, the variance terms $E_{\boldsymbol{\epsilon}}\langle \boldsymbol{w}, \boldsymbol{v}\rangle$ and $E_{\boldsymbol{\epsilon}}\langle \boldsymbol{v}, \boldsymbol{v}\rangle$ are required. In this section, we will show how to calculate these terms in linear regression with a quadratic regularizer $R(\theta) = \hat{\boldsymbol{\theta}}^T R\hat{\boldsymbol{\theta}}$, which results in the original SIC [10, 29, 18].

Let K be the $n \times p$ matrix whose (i, j) element is $\phi_j(x_i)$ and $\boldsymbol{y} = (y_1, \ldots, y_n)^T$. K is sometimes called the *design matrix* [30]. When $(\frac{1}{n}K^TK + \lambda R)$ is invertible, $\hat{\boldsymbol{\theta}}$ is given as [18]:

$$\hat{\boldsymbol{\theta}} = \frac{1}{n}(\frac{1}{n}K^TK + \lambda R)^{-1}K^Ty. \tag{19}$$

When K^TK is invertible, an unbiased estimate $\hat{\boldsymbol{\theta}^u}$ is given as [10]

$$\hat{\boldsymbol{\theta}^u} = (K^TK)^{-1}K^Ty. \tag{20}$$

Then the first term in (18) can be calculated. The second and third term can be exactly calculated as [10, 18]

$$E_{\boldsymbol{\epsilon}}\langle \boldsymbol{w}, \boldsymbol{v}\rangle = \sigma^2\mathrm{tr}(PW), \tag{21}$$
$$E_{\boldsymbol{\epsilon}}\langle \boldsymbol{v}, \boldsymbol{v}\rangle = \sigma^2\mathrm{tr}(PV), \tag{22}$$

where $\mathrm{tr}(\cdot)$ denotes the sum of diagonal elements of a matrix, where P is defined by (11), and W and V are the $p \times p$ matrices defined as

$$W = \frac{1}{n}(\frac{1}{n}K^TK + \lambda R)^{-1}, \tag{23}$$
$$V = (K^TK)^{-1}. \tag{24}$$

This finally yields the original SIC for quadratic regularizers [18]:

$$\mathrm{SIC} = (\hat{\boldsymbol{\theta}} - \hat{\boldsymbol{\theta}^u})^T P(\hat{\boldsymbol{\theta}} - \hat{\boldsymbol{\theta}^u}) + 2\sigma^2\mathrm{tr}(PW) - \sigma^2\mathrm{tr}(PV), \tag{25}$$

which gives an unbiased estimate of J_G [10]. Usually, the variance σ^2 is not known, so we use its unbiased estimate instead. When $n > p$, its unbiased estimate is given as [31]

$$\hat{\sigma}^2 = \frac{\boldsymbol{y}^T\boldsymbol{y} - (K\hat{\boldsymbol{\theta}^u})^T\boldsymbol{y}}{n - p}. \tag{26}$$

Even when we replace σ^2 by $\hat{\sigma}^2$, the unbiasedness property is conserved [10].

3.4 GSIC for Non-Quadratic Regularizers

When we are concerned with non-quadratic regularizers, $\hat{\boldsymbol{\theta}}$ can not be obtained analytically like in (19). Instead, it is usually obtained by some optimization method (e.g. [32, 8]). For this reason, it is difficult to evaluate the second term $E_{\boldsymbol{\epsilon}}\langle \boldsymbol{w}, \boldsymbol{v} \rangle$ in (18). So we approximate $E_{\boldsymbol{\epsilon}}\langle \boldsymbol{w}, \boldsymbol{v} \rangle$ under the assumption that the Hessian $H = [\frac{\partial^2 L_r}{\partial \theta_i \partial \theta_j}]$ of the loss function L_r is invertible for any $\hat{\boldsymbol{\theta}}$. Then, $E_{\boldsymbol{\epsilon}}\langle \boldsymbol{w}, \boldsymbol{v} \rangle$ is approximated as

$$E_{\boldsymbol{\epsilon}}\langle \boldsymbol{w}, \boldsymbol{v} \rangle \approx \sigma^2 \mathrm{tr}(PW^0), \tag{27}$$

where

$$W^0 = \frac{1}{n}\left(\frac{1}{n}K^T K + \frac{1}{2}\lambda \nabla\nabla R(\hat{\boldsymbol{\theta}})\right)^{-1}, \tag{28}$$

and $\nabla\nabla R(\hat{\boldsymbol{\theta}})$ is the $p \times p$ matrix whose (i,j) element is $\frac{\partial R(\boldsymbol{\theta})}{\partial \theta_i \partial \theta_j}\big|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}}$. The derivation of this approximation is described in appendix I. It gives GSIC for non-quadratic regularizers, which we propose in this paper:

Definition 2 (GSIC for Non-Quadratic Regularizers) *The following functional is called the Generalized Subspace Information Criterion for non-quadratic regularizers:*

$$\mathrm{GSIC} = (\hat{\boldsymbol{\theta}} - \hat{\boldsymbol{\theta}^u})^T P (\hat{\boldsymbol{\theta}} - \hat{\boldsymbol{\theta}^u}) + 2\sigma^2 \mathrm{tr}(PW^0) - \sigma^2 \mathrm{tr}(PV), \tag{29}$$

where $\hat{\boldsymbol{\theta}^u}$, P, W^0, and V are given by (20), (11), (28), and (24), respectively.

When the regularization term is quadratic, GSIC agrees with the original SIC (25). When σ^2 is not known, it is replaced by $\hat{\sigma}^2$ as in SIC. With regard to the relationship between GSIC and the generalization error J_G, we have the following theorem.

Theorem 1 *Assuming that $\hat{\boldsymbol{\theta}}$ can be represented as a b-th ($b < \infty$) order polynomial of $\boldsymbol{y}$ and the moments of ϵ_i up to b-th order are bounded, then GSIC for non-quadratic regularizers is an asymptotic unbiased estimate of J_G:*

$$E_{\boldsymbol{\epsilon}}[\mathrm{GSIC}] = J_G + O(n^{-2}). \tag{30}$$

A proof of the above theorem is provided in appendix II. In order to discuss theoretical properties of GSIC with more mathematical rigor, it would be necessary to obtain a version of the theorem without the assumption that $\hat{\boldsymbol{\theta}}$ is a finite order polynomial of $\boldsymbol{y}$. This, however, would go beyond the scope of this paper, which explores mainly the practical performance of GSIC.

4 Biased GSIC

In practical situations, it is common that as many basis functions as training examples are used, e.g. the Gaussian functions centered on all input points. In such cases, the unbiased solution $\hat{\boldsymbol{\theta}^u}$ tends to have a large variance, which also makes the variance of GSIC large. Therefore model selection can become unstable.

For reducing the variance, it is effective to replace $\hat{\boldsymbol{\theta}^u}$ by $\hat{\boldsymbol{\theta}^\alpha}$ obtained by weight decay regularization as

$$\hat{\boldsymbol{\theta}^\alpha} = (K^T K + \alpha I)^{-1} K^T \boldsymbol{y}, \tag{31}$$

where I is the $p \times p$ identity matrix. The (conceptual) distributions of $\hat{\boldsymbol{\theta}^u}$ and $\hat{\boldsymbol{\theta}^\alpha}$ are illustrated in Fig. 2. Although the mean of $\hat{\boldsymbol{\theta}^\alpha}$ has a small bias away from the true parameter $\boldsymbol{\theta}^*$, the variance of $\hat{\boldsymbol{\theta}^\alpha}$ becomes much smaller than that of $\hat{\boldsymbol{\theta}^u}$. We observe that by using the regularized $\hat{\boldsymbol{\theta}^\alpha}$ instead of $\hat{\boldsymbol{\theta}^u}$, GSIC becomes slightly biased but its variance is drastically reduced. However, now another regularization constant α has to be determined. By adjusting α such that $\hat{\boldsymbol{\theta}^\alpha}$ is an accurate estimator of $\boldsymbol{\theta}^*$, the error of GSIC is expected to be improved. Indeed, this expectation is supported by simulations in Sec.6.1. Fortunately, it is by far easier to determine α for weight decay regularization than to determine λ in the sparse regressor since in the weight decay case, the leave-one-out error can be efficiently computed in closed-form [5]:

$$\mathrm{LOOerror} = \frac{\boldsymbol{y}^T U (\mathrm{diag}(U))^{-2} U \boldsymbol{y}}{n}, \tag{32}$$

where $U = I_n - K(K^T K + \alpha I)^{-1} K^T$ and I_n denotes the $n \times n$ identity matrix. Also other sophisticated methods are available such as C_L [12] and GCV [33][5]. By using the closed-form result for the weight decay regularization parameter α, a good estimate of the noise variance σ^2 is obtained as (see e.g. [33])

$$\hat{\sigma}^2 = \frac{\boldsymbol{y}^T Z^2 \boldsymbol{y}}{\mathrm{tr}(Z)}, \tag{33}$$

where $Z = I - K(K^T K + \alpha I)^{-1} K^T$. Note that using (33) instead of (26) also slightly increases the bias of GSIC, but the variance is even further decreased. We call this technique *biased GSIC*(GSICb).

5 Applying GSIC to Sparse Regression

It is well-known that the ℓ_1 norm regularization leads to a sparse solution, where most of the parameters θ_i's are zero [19, 21]. A sparse regressor is practically useful because it

[5] A detailed review for weight decay regularization is available in [5].

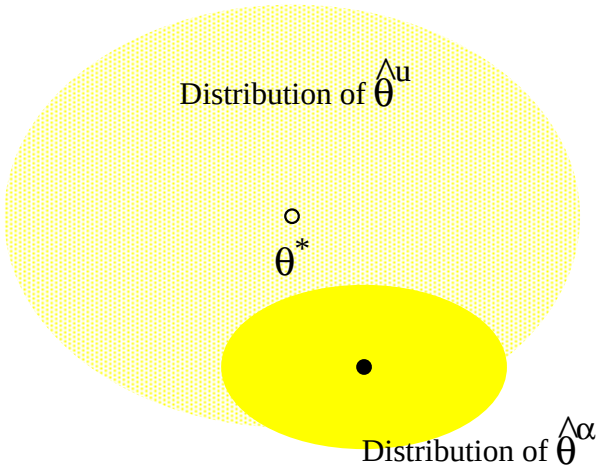


Figure 2: Illustration of the distributions of $\hat{\boldsymbol{\theta}}^u$ (not regularized) and $\hat{\boldsymbol{\theta}}^\alpha$ (regularized). The difference is that the variance of $\hat{\boldsymbol{\theta}}^\alpha$ gets smaller and the mean of $\hat{\boldsymbol{\theta}}^\alpha$ (denoted as $\bullet$ in the figure) does no longer coincide with the true parameter $\boldsymbol{\theta}^*$. The gain of shrinking the variance is expected to by far exceed this bias.

automatically selects necessary basis functions and moreover a sparse solution saves the computational cost. The loss function for the sparse regressor is given as

$$L_r = \frac{1}{n}\sum_{i=1}^{n}(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) - y_i)^2 + \lambda\sum_{i=1}^{p}|\theta_i|. \tag{34}$$

Minimizing L_r with respect to $\boldsymbol{\theta}$ is done by a convex quadratic programming [32]. Let us decompose $\boldsymbol{\theta} = \boldsymbol{\theta}^+ - \boldsymbol{\theta}^-$, where all elements of $\boldsymbol{\theta}^+$ and $\boldsymbol{\theta}^-$ are nonnegative. Then, the minimizer of L_r with respect to $\boldsymbol{\theta}$ is obtained by finding $\boldsymbol{\theta}^+$ and $\boldsymbol{\theta}^-$ that minimize

$$\frac{1}{n}\boldsymbol{\xi}^T\boldsymbol{\xi} + \lambda\sum_{i=1}^{p}(\theta_i^+ + \theta_i^-) \tag{35}$$

under the constraint that $K(\boldsymbol{\theta}^+ - \boldsymbol{\theta}^-) = \boldsymbol{y} + \boldsymbol{\xi}$, $\boldsymbol{\theta}^+ \geq 0$, and $\boldsymbol{\theta}^- \geq 0$. We briefly explain why solving the problem (35) leads to the optimal solution of (34): At the optimal solution, either θ_i^+ or θ_i^- is zero, because otherwise the value of (35) can be reduced without violating constraints by the following manipulation:

$$\begin{aligned}
\{\theta_i^+, \theta_i^-\} &\leftarrow \{\theta_i^+ - \theta_i^-, 0\} \quad \text{when } \theta_i^+ - \theta_i^- \geq 0, \\
\{\theta_i^+, \theta_i^-\} &\leftarrow \{0, \theta_i^- - \theta_i^+\} \quad \text{when } \theta_i^+ - \theta_i^- < 0.
\end{aligned}$$

When this implicit constraint is taken into account,

$$\theta_i^+ + \theta_i^- = |\theta_i|. \tag{36}$$

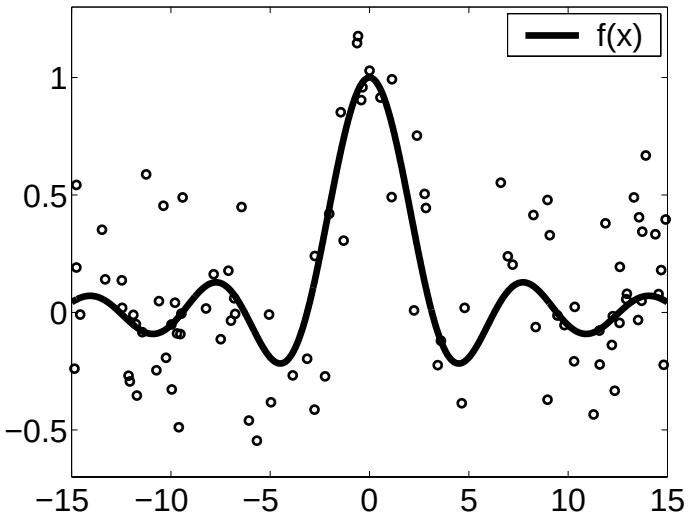


Figure 3: Learning target function and 100 training examples with $\sigma = 0.3$.

We can see that the two problems are equivalent because (34) is obtained by substituting (36) and the equality constraints into (35).

As for sparse regressors $\nabla\nabla R$ is not well defined, GSIC cannot applied directly. In order to apply GSIC to the sparse regressor, we need to approximate the regularization term $R(\boldsymbol{\theta}) = \sum_{i=1}^{p} |\theta_i|$ by a continuous function as

$$R'(\boldsymbol{\theta}) = \sum_{i=1}^{p} \theta_i \tanh(\gamma\theta_i), \tag{37}$$

where the slope is e.g. $\gamma = 10$. Then, $\nabla\nabla R$ is a diagonal matrix whose i-th element is

$$\nabla\nabla R_{ii} = 2(\gamma \mathrm{sech}^2(\gamma\hat{\theta}_i) - \gamma^2\hat{\theta}_i \mathrm{sech}^2(\gamma\hat{\theta}_i)\tanh(\gamma\hat{\theta}_i)). \tag{38}$$

Using (38), we can compute W^0 from Eq.(28) and therefore calculate GSIC for the sparse regressor. Because of this approximation, we are investigating the generalization error of the approximated regressor, not that of the sparse regressor itself. It is a further interesting topic to consider how to choose the approximator for minimizing the difference of the generalization errors, but outside the scope of this contribution.

6 Experiments

In this section, we perform experiments for sparse regressors. Notice that the purpose of the experiments is to demonstrate that GSIC actually works for generalization error prediction of sparse regressors.

6.1 Illustrative Example

Let the regression function be

$$f_{\boldsymbol{\theta}}(x) = \sum_{i=1}^{50} \theta_i \exp\left(-\frac{\|x - s_i\|^2}{\eta^2}\right), \tag{39}$$

where $\eta = 1$ and 50 template samples s_i's are equally spaced in $[-15, 15]$. We obtain the true parameter $\boldsymbol{\theta}^*$ by the least mean squares estimate with $\{(s_i, g(s_i))\}_{i=1}^{50}$, where

$$g(x) = |x|^{-1} \sin|x|. \tag{40}$$

For training, n input points $\{x_i\}_{i=1}^n$ are chosen randomly from the uniform distribution on $[-15, 15]$. The output values are obtained as $y_i = f(x_i) + \epsilon_i$, where ϵ_i's are independently subject to a normal distribution with mean zero and standard deviation σ. The target function and training examples are displayed in Fig. 3. The regularization constant is selected from

$$\lambda = 1.0 \times 10^{-4}, 1.0 \times 10^{-3.5}, \ldots, 1.0 \times 10^{-1} \tag{41}$$

by 10-fold cross validation (CV), NIC, GSIC and GSICb. Also, 100 additional unlabeled samples $\{x'_i\}_{i=1}^{100}$ are given from the uniform distribution on $[-15, 15]$. In GSIC and GSICb, the distribution $q(x)$ of these additional input points is estimated by the empirical distribution of the unlabeled samples:

$$q(x) = \frac{1}{100} \sum_{i=1}^{100} \delta(x - x'_i), \tag{42}$$

where $\delta(x) = 1$ when $x = 0$ and otherwise $\delta(x) = 0$. The true generalization error is measured by

$$\text{Error} = \int_{-15}^{15} \left(f_{\boldsymbol{\theta}}(x) - f(x)\right)^2 dx. \tag{43}$$

The performance of CV, NIC, and GSIC is measured by the generalization error at the selected λ (Fig. 4). The experiment consists of 100 trials with different noise. When $n = 200$, all criteria work well with no significant difference. As n decreases to 60, CV still works well, but NIC and GSIC tend to give a large generalization error.

In order to investigate the cause of errors by NIC and GSIC in detail, actual values of CV, NIC, and GSIC are displayed in Fig. 5 for $(n, \sigma) = (60, 0.3)$ and $(200, 0.3)$. Note that the values of GSIC in the figure are biased, because we ignored the terms $\|\hat{\boldsymbol{\theta}}^u\|^2$ and $\hat{\sigma}^2 \mathrm{tr}(PV)$, which are irrelevant to model selection. Thus we can see the essential contributions to the variance of the estimate.

When $(n, \sigma) = (200, 0.3)$, the shape of the curves by CV, NIC, and GSIC are very close to the true curve, which explains why the model selection was carried out successfully.

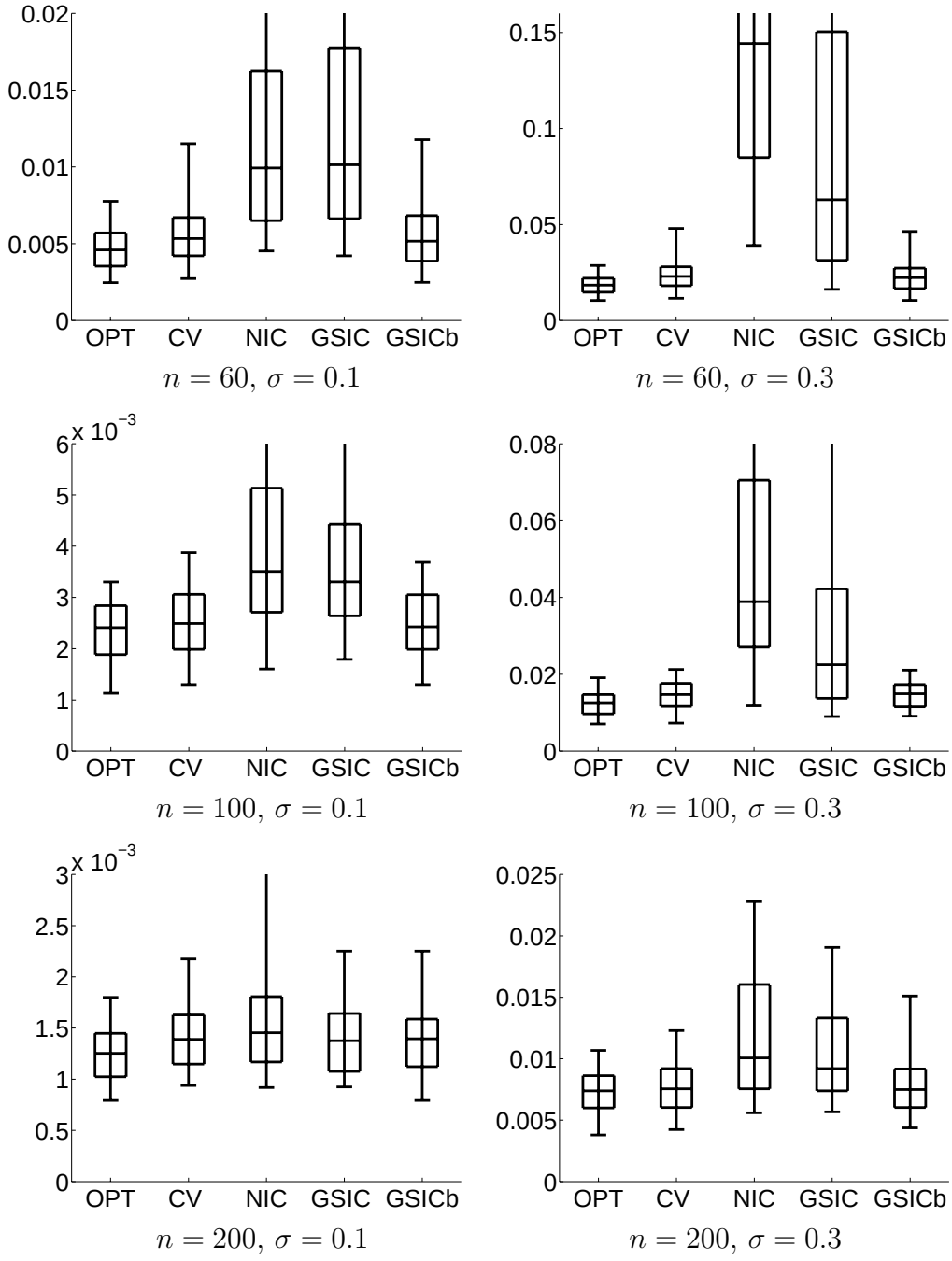
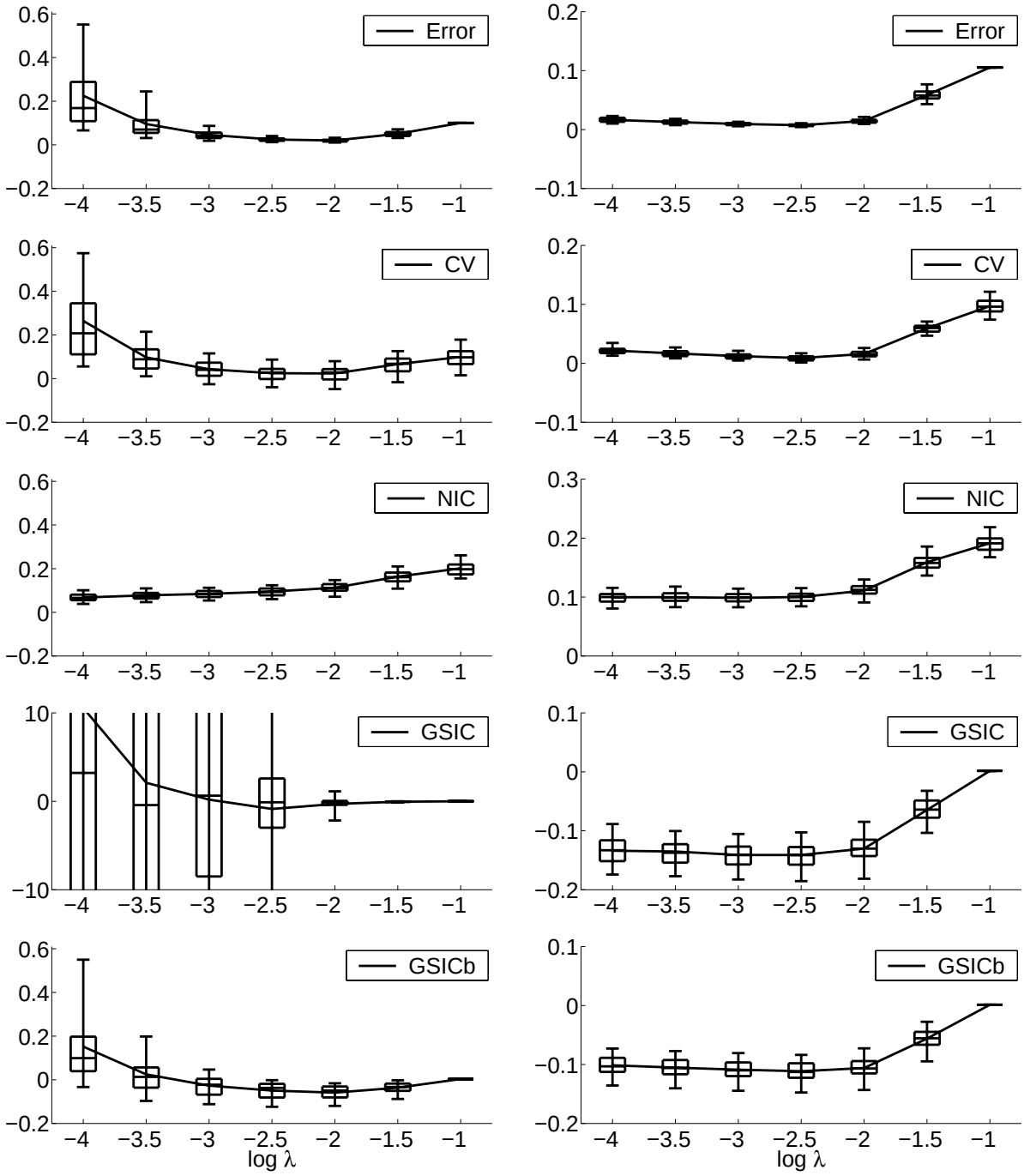


Figure 4: Generalization errors at selected λ for the respective model selection criterion shown with standard box plot (100 trials). The box plot notation specifies marks at 95, 75, 50, 25, and 5 percentiles of values. 'OPT' denotes the generalization error with the optimal λ.



$n = 60$, $\sigma = 0.3$ $n = 200$, $\sigma = 0.3$

Figure 5: Values of each criterion by 100 trials shown with standard box plot. The horizontal axis denotes $\log \lambda$. The solid line denotes the mean values.

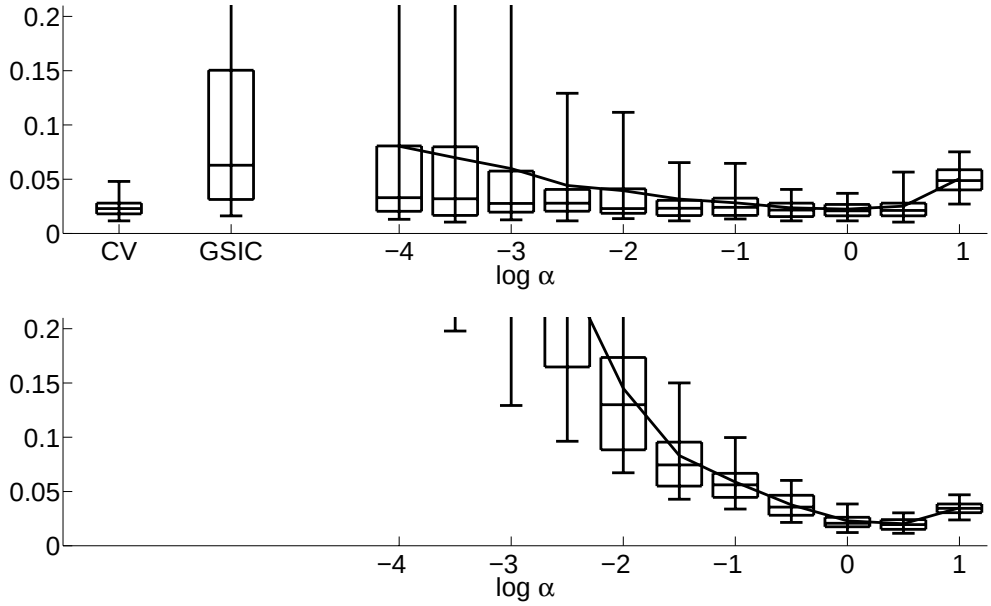


Figure 6: (Top) The generalization error at selected λ by GSICb with changing α ($n = 60$ and $\sigma = 0.3$). The horizontal axis denotes $\log \alpha$. The results of cross validation and GSIC are also shown for comparison. (Bottom) Generalization errors of $\hat{\boldsymbol{\theta}}^{\alpha}$ with changing α.

Although CV still gives an accurate curve when $(n, \sigma) = (60, 0.3)$, the curves of NIC and GSIC are no longer accurate. These graphs also show that the inaccuracy of the curves by NIC and GSIC has different characteristics. The NIC curve is tilted towards the left, which shows that NIC tends to choose smaller regularization constants. This figure tells us that the unbiasedness of NIC is essentially lost because of the small sample effect. In GSIC, huge variance dominates the graph, so the shape of the average curve is unreliable. The variance of GSIC is large especially when the regularization constant λ is small. So, for explaining the failure in NIC, the bias plays a main role whereas in GSIC, the variance is of primal importance.

In order to reduce the variance of GSIC, we use the biased version GSICb. The top graph in Fig. 6 shows the generalization error of GSICb at the selected λ value as a function of the weight decay parameter α:

$$\alpha = 1.0 \times 10^{-4}, 1.0 \times 10^{-3.5}, \ldots, 1.0 \times 10^{1}. \tag{44}$$

The bottom graph in Fig. 6 displays the true generalization error of $\hat{\boldsymbol{\theta}}^{\alpha}$ with changing α. These graph shows that the minimum of these two curves approximately agrees. This means that if α is determined such that the true generalization error of $\hat{\boldsymbol{\theta}}^{\alpha}$ is minimized, then the performance of GSICb is expected to be the best. In the experiments, we use a leave-one-out cross-validation to approximate the true generalization error and thus to determine α (see Sec. 4). Note that for GSICb, the noise variance is estimated by

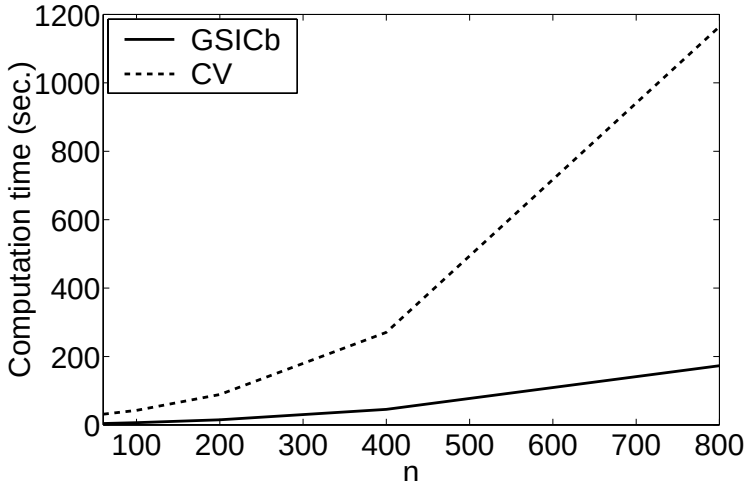


Figure 7: Computation time. The horizontal axis denotes the number of training examples and the vertical axis denotes the computation time in seconds. The plot of GSICb shows the overall computation time which includes the model selection procedure for choosing α. The plot of CV shows the computation time of actually performing the 10-fold cross validation procedure by repeating to solve the quadratic programming problem (35).

(33). Fig. 4 shows that GSICb works as well as other methods when $n = 200$. With the decrease of n to 60, GSICb tends to work much better than NIC and non-regularized GSIC, and its performance is comparable to CV. Fig. 5 shows that the shape of the GSICb curve shadows the true curve nicely when $(n, \sigma) = (200, 0.3)$. Note that terms which are irrelevant to model selection are ignored also in GSICb because of the similar reason to above. When $(n, \sigma) = (60, 0.3)$, the variance of the GSICb curve is far reduced compared to that of the non-regularized GSIC curve, and its shape coincides very well with the true curve. This implies that the introduction of regularization parameter α for obtaining a reference estimator (cf. (31) and (33)) drastically reduces the variance with an irrelevant effect on the bias. Therefore, GSICb works well even for small samples.

The computation times of GSICb and CV are plotted in Fig. 7. The plot of GSICb shows the overall computation time which includes the model selection procedure for choosing α. The plot of CV shows the computation time of actually performing the 10-fold cross validation procedure by repeating to solve the convex quadratic programming problem (35). In this experiment GSICb is much faster than CV, and the advantage increases as n becomes larger.[6]

In summary, this illustrative one dimensional experiment shows that non-regularized GSIC performs well when n is large, but it can become unstable for small sample cases. Although it is heuristically derived, GSICb works comparably well as CV in the cases studied and it is computationally much more efficient than CV.

[6]The computation of the leave-one-out error for weight decay (32) is $O(n^2)$.

6.2 Experiment on Multidimensional Data

To further inspect the performance of GSIC(b), we studied a number of multidimensional data sets provided by DELVE [34]; we will report exemplarily about results on the Boston Housing data in this work. The Boston Housing data set has 506 points in 14 dimensional space, where we used the 14th variable MEDV as the output value. Each input variable is divided by its maximum value for normalization. We randomly choose 50 samples for training and 100 samples as unlabeled data. The 356 remaining test samples are used for measuring the generalization error. The regression function is described as

$$f_{\boldsymbol{\theta}}(x) = \sum_{i=1}^{50} \theta_i k(\boldsymbol{x}, \boldsymbol{x}_i), \tag{45}$$

where k is the third-order ANOVA decomposition kernel [35, 1]:

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_{1 \leq k_1 < k_2 < \ldots < k_3 \leq 13} \varphi(x_{ik_1}, x_{jk_1}) \varphi(x_{ik_2}, x_{jk_2}) \varphi(x_{ik_3}, x_{jk_3}) \tag{46}$$

constructed from a linear spline kernel φ [36]:

$$\varphi(x_i, x_j) = 1 + x_i x_j + x_i x_j \min(x_i, x_j) - \frac{x_i + x_j}{2} (\min(x_i, x_j))^2 + \frac{(\min(x_i, x_j))^3}{3}. \tag{47}$$

Here, all of the 50 training samples are used as template samples. As candidate values for the regularization parameter λ, 10 equally spaced points in the log scale are taken from $[10^{-3}, 10^3]$. In GSICb, we used the same candidate values for α. Note that, in the cross validation process of this experiment, the basis functions corresponding to hold-out training samples are not used, i.e. the regressor from Eq.(45) has accordingly fewer basis functions. The result of 100 trials are summarized in Fig. 8. We included the result of leave-one-out cross validation as well as 10-fold cross validation. Even in the challenging situation that the number of samples is the same as the number of parameters, GSICb performed comparably to 10-fold cross validation and leave-one-out cross validation.

7 Concluding Remarks

In this paper, we proposed GSIC and GSICb, two generalization error prediction methods for non-quadratic regularizers, which make use of the distribution of additional unlabeled input points. They extend SIC, whose range of application is limited to quadratic regularizers. Theoretically, the bias of GSIC was shown to vanish asymptotically. In experiments, GSIC worked well with larger samples in its original form, and its regularized variant GSICb worked excellently even for small sample sizes. Therefore GSIC(b) is an interesting *stand-alone* model selection technique. Another aspect of GSICb is that it makes use of a well-tuned reference estimator. So conceptually, we can understand GSICb as a technique to achieve good model selection from a reference estimator, i.e. we

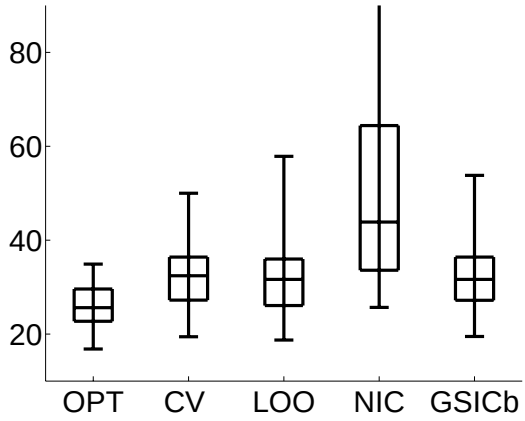


Figure 8: Generalization errors at selected λ by 100 trials in Boston Housing dataset. The number of samples is 50. The results are shown with the standard box plot. 'LOO' denotes the result of leave-one-out cross validation

can *transfer* regularization knowledge from one learning machine to another. GSICb is especially useful when the model selection of the reference estimator – as in our case – can be done efficiently. Thus we can save a good amount of computation time.

Future work will focus on theoretical aspects of choosing reference estimators for GSICb. An interesting question here is how to *optimally* transfer e.g. regularization information from a reference estimator to another learning machine or in general between two learning machines. For this purpose, we will analyze both bias and variance of GSICb, for instance along the lines of [37]. Furthermore we plan to apply GSIC(b) to classification and unsupervised learning.

Acknowledgements K.R.M gratefully acknowledges partial financial support from DFG under contracts JA 379/91, MU 987/11 and the EU in the Neurocolt 2 and the BLISS project (IST-1999-14190). We thank Shun-ichi Amari, Hidemitsu Ogawa, Takashi Onoda, Gunnar Rätsch and Sebastian Mika for valuable discussions. Special thanks goes to the anonymous referees who strongly helped to improve this paper. MS would like to thank GMD for warm hospitality.

A Derivation of GSIC

In this section, we will show the derivation of (27). Because we assumed that the solution of the learning problem is unique, $\hat{\boldsymbol{\theta}}$ is considered as a function of $\boldsymbol{y}$. Let $\boldsymbol{z} := (f(x_1), \ldots, f(x_n))^T$ and $\boldsymbol{\epsilon} := (\epsilon_1, \ldots, \epsilon_n)^T$. Also, let the derivatives of $\hat{\boldsymbol{\theta}}(\boldsymbol{y})$ be

denoted as

$$\nabla\hat{\theta}_i(\boldsymbol{y}) := \left(\frac{\partial\hat{\theta}_i}{\partial y_1}(\boldsymbol{y}), \ldots, \frac{\partial\hat{\theta}_i}{\partial y_n}(\boldsymbol{y})\right)^T, \tag{48}$$

and $\nabla\hat{\boldsymbol{\theta}}(\boldsymbol{y}) := (\nabla\hat{\theta}_1(\boldsymbol{y}), \cdots, \nabla\hat{\theta}_p(\boldsymbol{y}))^T$. Then, $\hat{\boldsymbol{\theta}}(\boldsymbol{y})$ can be expressed via Taylor expansion as follows:

$$\hat{\theta}_i(\boldsymbol{y}) = \hat{\theta}_i(\boldsymbol{z}+\boldsymbol{\epsilon}) = \hat{\theta}_i(\boldsymbol{z}) + \nabla\hat{\theta}_i(\boldsymbol{z})^T\boldsymbol{\epsilon} + S_i, \tag{49}$$

where S_i is the residual. Then, w_i (i-th element of $\boldsymbol{w}$) is described as

$$w_i = \hat{\theta}_i(\boldsymbol{z}+\boldsymbol{\epsilon}) - E_{\boldsymbol{\epsilon}}[\hat{\theta}_i(\boldsymbol{z}+\boldsymbol{\epsilon})] = \nabla\hat{\theta}_i(\boldsymbol{z})^T\boldsymbol{\epsilon} + S_i - E_{\boldsymbol{\epsilon}}[S_i]. \tag{50}$$

Expressing an unbiased estimator $\hat{\boldsymbol{\theta}^u}(\boldsymbol{y})$ by (20)[7], $\nabla\hat{\boldsymbol{\theta}^u}(\boldsymbol{y})$ is given as

$$\nabla\hat{\boldsymbol{\theta}^u} = (K^TK)^{-1}K^T, \tag{51}$$

and hence v_i (i-th element of $\boldsymbol{v}$) is described as

$$v_i = \nabla\hat{\theta}_i^{uT}\boldsymbol{\epsilon}. \tag{52}$$

Now $E_{\boldsymbol{\epsilon}}\langle\boldsymbol{w},\boldsymbol{v}\rangle$ is written as

$$E_{\boldsymbol{\epsilon}}\langle\boldsymbol{w},\boldsymbol{v}\rangle = \sum_{i=1}^{p}\sum_{j=1}^{p} P_{ij}E_{\boldsymbol{\epsilon}}[w_iv_j], \tag{53}$$

where $E_{\boldsymbol{\epsilon}}[w_iv_j]$ is expressed as

$$E_{\boldsymbol{\epsilon}}[w_iv_j] = \sigma^2\nabla\hat{\theta}_i(\boldsymbol{z})^T\nabla\hat{\theta}_j^u + E_{\boldsymbol{\epsilon}}[S_i(\nabla\hat{\theta}_j^{uT}\boldsymbol{\epsilon})]. \tag{54}$$

Here, we approximate $E_{\boldsymbol{\epsilon}}[w_iv_j]$ by

$$E_{\boldsymbol{\epsilon}}[w_iv_j] \approx \sigma^2\nabla\hat{\theta}_i(\boldsymbol{y})^T\nabla\hat{\theta}_j^u, \tag{55}$$

i.e., the second term of (54) is ignored and $\boldsymbol{z}$ in the first term is replaced by $\boldsymbol{y}$. The analysis of the error due to the approximation using Eq.(55) will be given in the proof of Theorem 1 (Appendix II) under several assumptions. Then we obtain

$$E_{\boldsymbol{\epsilon}}\langle\boldsymbol{w},\boldsymbol{v}\rangle \approx \sigma^2\mathrm{tr}(PW^0) \tag{56}$$

where

$$W^0 = \nabla\hat{\boldsymbol{\theta}}(\boldsymbol{y})\nabla\hat{\boldsymbol{\theta}^u}^T. \tag{57}$$

[7]For simplicity, we derive the GSIC for non-quadratic regularizers based on this unbiased estimator here. However, you can trace the same way when you use a different unbiased estimator.

The derivatives $\nabla\hat{\boldsymbol{\theta}}(\boldsymbol{y})$ can be obtained from the saddle point equation,

$$\frac{\partial L_r}{\partial \theta_i} = 0, \ (i = 1, \ldots, p). \tag{58}$$

Differentiating the above equation with respect to y_k, we have

$$\sum_{j=1}^{p} \frac{\partial^2 L_r}{\partial \theta_i \partial \theta_j} \frac{\partial \theta_j}{\partial y_k} + \frac{\partial^2 L_r}{\partial \theta_i \partial y_k} = 0.$$

In matrix representation,

$$H\nabla\hat{\boldsymbol{\theta}}(\boldsymbol{y}) + M = \mathbf{0},$$

where H is a $p \times p$ matrix whose (i,j) element is $H_{ij} = \frac{\partial^2 L_r}{\partial \theta_i \partial \theta_j}$, and M is a $p \times n$ matrix whose (i,j) element is $M_{ij} = \frac{\partial^2 L_r}{\partial \theta_i \partial y_j}$. When H is invertible, $\nabla\hat{\boldsymbol{\theta}}(\boldsymbol{y})$ is described as

$$\nabla\hat{\boldsymbol{\theta}}(\boldsymbol{y}) = -H^{-1}M. \tag{59}$$

Substituting (5) to (59), we have

$$\nabla\hat{\boldsymbol{\theta}}(\boldsymbol{y}) = \frac{1}{n}\left(\frac{1}{n}K^T K + \frac{1}{2}\lambda\nabla\nabla R(\hat{\boldsymbol{\theta}}(\boldsymbol{y}))\right)^{-1} K^T. \tag{60}$$

Consequently, (28) is derived by substituting (51) and (60) into (57).

B Proof of Theorem 1

Here, we shall show that the order of the error due to the approximation in (55) is $O(n^{-2})$:

$$E_{\boldsymbol{\epsilon}}[S_i(\nabla\hat{\theta}_j^{uT}\boldsymbol{\epsilon}) - \sigma^2(\nabla\hat{\theta}_i(\boldsymbol{y}) - \nabla\hat{\theta}_j(\boldsymbol{z}))^T\nabla\hat{\theta}_j^u] = O(n^{-2}) \tag{61}$$

First, we assume that $\hat{\boldsymbol{\theta}}(\boldsymbol{y})$ can be represented by b-th order polynomial ($b < \infty$), and the moments of ϵ_i up to b-th order are bounded. Then, $S_i = \sum_{a=2}^{b} S_{ia}$ and

$$S_{ia} = \frac{1}{a!}\sum_{k_1=1}^{n}\cdots\sum_{k_a=1}^{n}\frac{\partial^a\hat{\theta}_i}{\partial y_{k_1}\cdots\partial y_{k_a}}(\boldsymbol{z})\epsilon_{k_1}\cdots\epsilon_{k_a}. \tag{62}$$

We first show the following lemmas.

Lemma 1 *Let $i_1, \ldots, i_a$ denote a set of indices such that $1 \leq i_1, \ldots, i_a \leq n$. Then, the following relation holds:*

$$E_{\boldsymbol{\epsilon}}[\sum_{i_1=1}^{n}\cdots\sum_{i_a=1}^{n}\prod_{k=1}^{a}\epsilon_{i_k}] = \begin{cases} 0 & (\ a \text{ is odd }\ ) \\ O(n^{a/2}) & (a \text{ is even}) \end{cases} \tag{63}$$

Lemma 2 *The order of a-th order derivatives of* $\hat{\boldsymbol{\theta}}(\boldsymbol{y})$ *is described as*

$$\frac{\partial^a \hat{\theta}_i}{\partial y_{i_1} \cdots \partial y_{i_a}} = O(n^{-a}). \tag{64}$$

Proofs of the above lemmas are given in appendix III and IV. Note that Lemma 2 also holds for $\hat{\boldsymbol{\theta}}^u(\boldsymbol{y})$ by setting $\lambda = 0$.

First, we will derive the order of the term $E_{\boldsymbol{\epsilon}}[S_i(\nabla\hat{\theta}_j^{uT}\boldsymbol{\epsilon})]$.

$$\begin{aligned} E_{\boldsymbol{\epsilon}}[S_{ia}(\nabla\hat{\theta}_j^{uT}\boldsymbol{\epsilon})] &= E_{\boldsymbol{\epsilon}}\left(\sum_{k=1}^n \frac{\partial\hat{\theta}_j^u}{\partial y_k}\epsilon_k\right)\left(\frac{1}{a!}\sum_{i_1=1}^n\cdots\sum_{i_a=1}^n \frac{\partial^a\hat{\theta}_i}{\partial y_{i_1}\cdots\partial y_{i_a}}\epsilon_{i_1}\cdots\epsilon_{i_a}\right) \\ &= \frac{1}{a!}\sum_{k=1}^n\sum_{i_1=1}^n\cdots\sum_{i_a=1}^n \frac{\partial\hat{\theta}_j^u}{\partial y_k}\frac{\partial^a\hat{\theta}_i}{\partial y_{i_1}\cdots\partial y_{i_a}}E_{\boldsymbol{\epsilon}}[\epsilon_k\epsilon_{i_1}\cdots\epsilon_{i_a}] \end{aligned}$$

By using Lemmas 1 and 2, we have $E_{\boldsymbol{\epsilon}}[S_{ia}(\nabla\hat{\theta}_j^{uT}\boldsymbol{\epsilon})] = O(n^{-1-a+(a+1)/2})$ when a is odd and 0 when a is even. Then, the order of $E_{\boldsymbol{\epsilon}}[S_i(\nabla\hat{\theta}_j^{uT}\boldsymbol{\epsilon})]$ is equal to that of its leading term $E_{\boldsymbol{\epsilon}}[S_{i3}(\nabla\hat{\theta}_j^{uT}\boldsymbol{\epsilon})] = O(n^{-2})$.

Next, we focus on the other term $E_{\boldsymbol{\epsilon}}[(\nabla\hat{\theta}_i(\boldsymbol{y}) - \nabla\hat{\theta}_j(\boldsymbol{z}))^T\nabla\hat{\theta}_j^u]$, which is described as

$$E_{\boldsymbol{\epsilon}}[(\nabla\hat{\theta}_i(\boldsymbol{y}) - \nabla\hat{\theta}_j(\boldsymbol{z}))^T\nabla\hat{\theta}_j^u] = \sum_{k=1}^n \frac{\partial\hat{\theta}_j^u}{\partial y_k}(E_{\boldsymbol{\epsilon}}[\frac{\partial\hat{\theta}_i}{\partial y_k}(\boldsymbol{z}+\boldsymbol{\epsilon})] - \frac{\partial\hat{\theta}_i}{\partial y_k}(\boldsymbol{z})). \tag{65}$$

By Taylor expansion, we have

$$E_{\boldsymbol{\epsilon}}[\frac{\partial\hat{\theta}_i}{\partial y_k}(\boldsymbol{z}+\boldsymbol{\epsilon})] - \frac{\partial\hat{\theta}_i}{\partial y_k}(\boldsymbol{z}) = E_{\boldsymbol{\epsilon}}[\sum_{l=1}^n \frac{\partial^2\hat{\theta}_i}{\partial y_k\partial y_l}(\boldsymbol{z})\epsilon_l] + E_{\boldsymbol{\epsilon}}[\sum_{a=2}^{b-1} T_a], \tag{66}$$

where

$$T_a = \frac{1}{a!}\sum_{i_1=1}^n\cdots\sum_{i_a=1}^n \frac{\partial^{a+1}\hat{\theta}_i}{\partial y_k\partial y_{i_1}\cdots\partial y_{i_a}}(\boldsymbol{z})\epsilon_{i_1}\cdots\epsilon_{i_a}. \tag{67}$$

The first term of the right-hand side of (66) is zero because each ϵ_i is independent and has zero mean. From Lemmas 1 and 2, we have $E_{\boldsymbol{\epsilon}}[T_a] = O(n^{-(a+1)+a/2})$, when a is even and 0 when a is odd. Then, the order of $E_{\boldsymbol{\epsilon}}[\sum_{a=2}^{b-1} T_a]$ is equal to that of its leading term $E_{\boldsymbol{\epsilon}}[\sum_{a=2}^{b-1} T_a] = O(n^{-2})$. By substituting this and the relation $\frac{\partial\hat{\theta}_j^u}{\partial y_k} = O(n^{-1})$ into (65), we have (61). ■

C Proof of Lemma 1

The task is to derive the order of Q_a:

$$Q_a = \sum_{i_1=1}^n\cdots\sum_{i_a=1}^n E_{\boldsymbol{\epsilon}}[\epsilon_{i_1}\cdots\epsilon_{i_a}], \tag{68}$$

Let us define an index vector $\boldsymbol{i} = (i_1, \ldots, i_a)^T$ and assume that $\boldsymbol{i}$ contains $r(\boldsymbol{i})$ unique values $v_1(\boldsymbol{i}) < \cdots < v_{r(\boldsymbol{i})}(\boldsymbol{i})$. Let the number of indices whose values are $v_j(\boldsymbol{i})$ be denoted as $m_j(\boldsymbol{i})(\geq 1)$:

$$m_j(\boldsymbol{i}) = |\{k \mid i_k = v_j(\boldsymbol{i})\}| \tag{69}$$

where $|\cdot|$ denote the cardinality of a set. For example, if $\boldsymbol{i} = (4,4,5,2,4,2)^T$, then $r(\boldsymbol{i}) = 3$, $v_1(\boldsymbol{i}) = 2$, $v_2(\boldsymbol{i}) = 4$, $v_3(\boldsymbol{i}) = 5$, $m_1(\boldsymbol{i}) = 2$, $m_2(\boldsymbol{i}) = 3$, and $m_3(\boldsymbol{i}) = 1$. Also, define the set of all index vectors as I, then Q_a can be rewritten as

$$Q_a = \sum_{\boldsymbol{i} \in I} E_{\boldsymbol{\epsilon}}[\epsilon_{i_1} \cdots \epsilon_{i_a}], \tag{70}$$

Let us define the subset of I as

$$I' = \{\boldsymbol{i} \mid m_j(\boldsymbol{i}) \text{ is even for } j = 1, \ldots, r(\boldsymbol{i})\}. \tag{71}$$

This excludes the index vectors where the same number appears odd times. $E_\epsilon(\epsilon_i^k) = 0$ when k is odd, because the distribution of ϵ_i is symmetric (See definition in Sec. 2) and ϵ_i's are independent. Therefore, we have $E_{\boldsymbol{\epsilon}}[\epsilon_{i_1} \cdots \epsilon_{i_a}] = 0$ for any $\boldsymbol{i} \in I'$. So, Q_a is rewritten as

$$Q_a = \sum_{\boldsymbol{i} \in I'} E_{\boldsymbol{\epsilon}}[\epsilon_{i_1} \cdots \epsilon_{i_a}]. \tag{72}$$

Since the noise moments are bounded by assumption, there exists a positive constant M that $E_{\boldsymbol{\epsilon}}[\epsilon_{i_1} \cdots \epsilon_{i_a}] \leq M$. Then, we have

$$Q_a \leq M|I'|. \tag{73}$$

When a is odd, at least one $m_j(\boldsymbol{i})$ is odd for any $\boldsymbol{i} \in I$. Therefore, I' is a null set and $Q_a = 0$. When a is even, $m_j(\boldsymbol{i}) \geq 2$ for all $\boldsymbol{i} \in I'$. Then, the number of unique values are bounded as $r(\boldsymbol{i}) \leq a/2$. So the cardinality of I' is less than $n^{a/2}$ and thus $|I'| = O(n^{a/2})$. ■

D Proof of Lemma 2

Let us describe that the a-th order derivative of $\hat{\theta}_k$ as

$$q_i^{(a)}(\hat{\boldsymbol{\theta}}(\boldsymbol{y}), \boldsymbol{y}) = \frac{\partial^a \hat{\theta}_k}{\partial y_{i_1} \cdots \partial y_{i_a}} \tag{74}$$

where $i_1, \ldots, i_a$ are indices in $[1, n]$. As a step to prove (64), we will prove that $q_i^{(a)}(\hat{\boldsymbol{\theta}}(\boldsymbol{y}), \boldsymbol{y})$ is described only by $\hat{\boldsymbol{\theta}}(\boldsymbol{y})$, that is, there is some function f such that $q_i^{(a)}(\hat{\boldsymbol{\theta}}(\boldsymbol{y}), \boldsymbol{y}) = f(\hat{\boldsymbol{\theta}}(\boldsymbol{y}))$.

We construct this proof by induction. When $c = 1$, it is obvious that $q_i^{(1)}$ is described only by $\hat{\boldsymbol{\theta}}(\boldsymbol{y})$ from (60). Assume that the c-th order derivative is described only by $\hat{\boldsymbol{\theta}}(\boldsymbol{y})$, that is, there is some function f such that $q_i^{(c)}(\hat{\boldsymbol{\theta}}(\boldsymbol{y}), \boldsymbol{y}) = f(\hat{\boldsymbol{\theta}}(\boldsymbol{y}))$. Then, the $(c+1)$-th order derivative $q_i^{(c+1)}(\hat{\boldsymbol{\theta}}(\boldsymbol{y}), \boldsymbol{y})$ can also be described by $\hat{\boldsymbol{\theta}}(\boldsymbol{y})$ only, because

$$q_i^{(c+1)}(\hat{\boldsymbol{\theta}}, \boldsymbol{y}) = \frac{\partial f}{\partial y_{i_{c+1}}} = \sum_{k=1}^{p} \frac{\partial f}{\partial \hat{\theta}_k} \frac{\partial \hat{\theta}_k}{\partial y_{i_{c+1}}} \tag{75}$$

and $\frac{\partial \hat{\theta}_k}{\partial y_{i_{c+1}}}$ is described only by $\hat{\boldsymbol{\theta}}(\boldsymbol{y})$ as in (60). By induction, it is proved that the derivatives of any order are described only by $\hat{\boldsymbol{\theta}}(\boldsymbol{y})$.

Then, we proceed to prove (64). When $c = 1$, it is obvious that $\frac{\partial \hat{\theta}_k}{\partial y_i} = O(n^{-1})$ from (60). Assume that the order of $q_i^{(c)}$ be $O(n^{-m})$, then the order of $q_i^{(c+1)}$ is $O(n^{-(m+1)})$ from (75). Therefore, by induction, we have proven (64) for any a. ■

References

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

[2] C.M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.

[3] V. Cherkassky and F. Mulier, *Learning from Data*, Wiley, New York, 1998.

[4] T. Poggio and F. Girosi, "Regularization algorithms for learning that are equivalent to multilayer networks," *Science*, vol. 247, pp. 978–982, 1990.

[5] M.J.L. Orr, "Introduction to radial basis function networks," Tech. Rep., Centre for Cognitive Science, University of Edinburgh, 1996.

[6] G. Orr and K.-R. Müller, Eds., *Neural Networks: Tricks of the Trade*, vol. 1524, Springer LNCS, 1998.

[7] V. Cherkassky, X. Shao, F.M. Mulier, and V.N. Vapnik, "Model complexity control for regression using VC generalization bounds," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 1075–1089, 1999.

[8] N. Murata, S. Yoshizawa, and S. Amari, "Network information criterion — determining the number of hidden units for an artificial neural network model," *IEEE Transactions on Neural Networks*, vol. 5, pp. 865–872, 1994.

[9] N. Murata, "Bias of estimators and regularization terms," in *Proceedings of 1998 Workshop on Information-Based Induction Sciences (IBIS'98)*, Izu, Japan, July 11-12 1998, pp. 87–94.

[10] M. Sugiyama and H. Ogawa, "Subspace information criterion for model selection," *Neural Computation*, vol. 13, no. 8, 2001.

[11] F. Mosteller and D. Wallace, "Inference in an authorship problem. a comparative study of discrimination methods applied to the authorship of the disputed Federalist papers," *Journal of the American Statistical Association*, vol. 58, pp. 275–309, 1963.

[12] C.L. Mallows, "Some comments on C_P," *Technometrics*, vol. 15, no. 4, pp. 661–675, 1973.

[13] H. Akaike, "Likelihood and the Bayes procedure," in *Bayesian Statistics*, N. J. Bernardo, M. H. DeGroot, D. V. Lindley, and A. F. M. Smith, Eds., Valencia, 1980, pp. 141–166, University Press.

[14] D.J.C. MacKay, "Bayesian interpolation," *Neural Computation*, vol. 4, no. 3, pp. 415–447, 1992.

[15] H. Akaike, "A new look at the statistical model identification," *IEEE Transactions on Automatic Control*, vol. AC-19, no. 6, pp. 716–723, 1974.

[16] D. Schuurmans and F. Southey, "An adaptive regularization criterion for supervised learning," *Proceedings of ICML'2000*, pp. 847–854, 2000.

[17] J. Hertz, A. Krogh, and R.G. Palmer, *Introduction to the Theory of Neural Computation*, Addison-Wesley, 1991.

[18] M. Sugiyama and H. Ogawa, "Optimal design of regularization term and regularization parameter by subspace information criterion," Technical Report TR00-0013, Department of Computer Science, Tokyo Institute of Technology, Tokyo, Japan, 2000.

[19] P.M. Williams, "Bayesian regularisation and pruning using a Laplace prior," *Neural Computation*, vol. 7, no. 1, pp. 117–143, 1995.

[20] K.P. Bennett and O.L. Mangasarian, "Robust linear programming discrimination of two linearly inseparable sets," *Optimization Methods and Software*, vol. 1, pp. 23–34, 1992.

[21] O.L. Mangasarian, "Mathematical programming in data mining," *Data Mining and Knowledge Discovery*, vol. 42, no. 1, pp. 183–201, 1997.

[22] P.S. Bradley, O.L. Mangasarian, and W.N. Street, "Feature selection via mathematical programming," *INFORMS Journal on Computing*, vol. 10, pp. 209–217, 1998.

[23] T. Graepel, R. Herbrich, B. Schölkopf, A.J. Smola, P.L. Bartlett, K.-R. Müller, K. Obermayer, and R.C. Williamson, "Classification on proximity data with LP-machines," in *Proceedings of ICANN'99*, D. Willshaw and A. Murray, Eds. 1999, vol. 1, pp. 304–309, IEE Press.

[24] A.J. Smola, O.L. Mangasarian, and B. Schölkopf, "Sparse kernel feature analysis," Tech. Rep. 99-04, University of Wisconsin, Data Mining Institute, Madison, 1999.

[25] A.J. Smola and B. Schölkopf, "Sparse greedy matrix approximation for machine learning," in *Proceedings of ICML'2000*, 2000, pp. 911–918.

[26] A.K. Jain, R.P.W. Duin, and J. Mao, "Statistical pattern recognition: A review," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 22, no. 1, pp. 4–37, 2000.

[27] S. Geman, E. Bienenstock, and R. Doursat, "Neural networks and the bias/variance dilemma," *Neural Computation*, vol. 4, no. 1, pp. 1–58, 1992.

[28] T. Heskes, "Bias/variance decompositions for likelihood-based estimators," *Neural Computation*, vol. 10, no. 6, pp. 1425–1433, 1998.

[29] M. Sugiyama and H. Ogawa, "Theoretical and experimental evaluation of subspace information criterion," *Machine Learning, Special Issue on New Methods for Model Selection and Model Combination*, 2001, to appear.

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

[31] V.V. Fedorov, *Theory of Optimal Experiments*, Academic Press, New York, 1972.

[32] S. Mika, G. Rätsch, and K.-R. Müller, "A mathematical programming approach to the kernel fisher algorithm," to appear in Neural Information Processing Systems 13, 2001.

[33] G. Wahba, *Spline Model for Observational Data*, vol. 59 of *Series in Applied Mathematics*, SIAM: Society for Industrial and Applied Mathematics, Philadelphia and Pennsylvania, 1990.

[34] University of Toronto, "`http://www.cs.utoronto.ca/~delve/data/datasets.html`," DELVE-Benchmark repository – a collection of artificial and real-world data sets.

[35] M. Stitson, A. Gammerman, V.N. Vapnik, V. Vovk, C. Watkins, and J. Weston, "Support vector regression with anova decomposition kernels," in *Advances in Kernel Methods — Support Vector Learning*, B. Schölkopf, C.J.C. Burges, and A.J. Smola, Eds., pp. 285–291. MIT Press, Cambridge, MA, 1999.

[36] V. Vapnik, S.W. Golowich, and A. Smola, "Support vector method for function approximation, regression estimation, and signal processing," *Neural Information Processing Systems 9*, pp. 281–287, 1997.

[37] H. Shimodaira, "An application of multiple comparison techniques to model selection," *Annals of Institute of Statistical Mathematics*, vol. 50, no. 1, pp. 1–13, 1998.