Input-Dependent Estimation of Generalization Error under Covariate Shift

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

Klaus-Robert Müller (klaus@first.fhg.de)
Fraunhofer FIRST, IDA, Kekuléstr. 7, 12489 Berlin, and
Department of Computer Science, University of Potsdam
August-Bebel-Str.89, Haus 4, 14482 Potsdam, Germany

Abstract

A common assumption in supervised learning is that the training and test input points follow the *same* probability distribution. However, this assumption is not fulfilled, e.g., in interpolation, extrapolation, active learning, or classification with imbalanced data. The violation of this assumption—known as the *covariate shift*—causes a heavy bias in standard generalization error estimation schemes such as cross-validation or Akaike's information criterion, and thus they result in poor model selection. In this paper, we propose an alternative estimator of the generalization error for the squared loss function when training and test distributions are different. The proposed generalization error estimator is shown to be exactly unbiased for finite samples if the learning target function is realizable and asymptotically unbiased in general. We also show that, in addition to the unbiasedness, the proposed generalization error estimator can accurately estimate the *difference* of the generalization error among different models, which is a desirable property in model selection. Numerical studies show that the proposed method compares favorably with existing model selection methods in regression for extrapolation and in classification with imbalanced data.

Keywords

linear regression, generalization error, model selection, covariate shift, sample selection bias, interpolation, extrapolation, active learning, classification with imbalanced data

AMS 1991 subject classification: Primary: 62J05; Secondary: 62F12

1 Introduction

It is most commonly assumed in supervised learning that the training and test *input* points follow the *same* probability distribution [50, 49, 13, 32]. However, this assumption is not fulfilled, for example, in *interpolation* or *extrapolation* scenarios: only few (or no) training input points exist in the regions of interest, implying that the test distribution is significantly different from the training distribution. *Active learning* also corresponds to such cases because the locations of training input points are designed by users while test input points are provided from the environment [9, 25, 28, 7, 11, 41, 42, 51, 19]. Another example is *classification with imbalanced data*, where the ratio of samples in each category is different between training and test phases.

The situation where the training and test distributions are different is referred to as the situation under the *covariate shift* [35] or the *sample selection bias* [14]. In such cases, two difficulties arise in a learning process. The first difficulty is parameter learning. The standard maximum likelihood estimation (MLE) tries to fit the data well in the region with high training data density, implying that the prediction can be inaccurate if the region with high test data density has low training data density. Theoretically, it is known that when the training and test distributions are different and the model is *misspecified* (i.e., the model can not express the learning target function), MLE is no longer *consistent* (i.e., the learned parameter does not converge to the optimal one even when the number of training examples goes to infinity)[1]. This problem can be overcome by using MLE weighted by the *ratio* of test and training data densities [35]. A key idea of this modification is that the training data density is adjusted to the test data density by the density ratio, which is similar in spirit to *importance sampling*. Although the consistency becomes guaranteed by this modification, the weighted version of MLE tends to have large variance. Indeed, it is no longer *asymptotically efficient* (i.e., its variance does not asymptotically attain the Cramér-Rao lower-bound). Therefore, in practical situations with finite samples, a stabilized estimator by means of, for example, changing the weight or adding a regularizer may be more appropriate. Thus, the parameter learning problem is now relocated to the model selection problem.

However, the second difficulty when the distributions of training and test input points are different is model selection itself. Standard unbiased generalization error estimation schemes such as *cross-validation* [24, 38, 50] or *Akaike's information criterion* [1, 46, 20, 17] are heavily biased, because the generalization error is over-estimated in the high training data density region and it is under-estimated in the high test data density region.

So far, there appear to be two attempts to cope with this problem. One attempt is an asymptotic statistical approach in the context of modifying AIC such that the asymptotic unbiasedness of AIC is still maintained even when the training and test distributions are different [35]. In the following, we refer to this method as the *modified AIC* (MAIC). A key idea of MAIC is again the use of the density ratio for compensating for the difference of training and test data densities. This approach assumes the availability of a large number

[1] Note, however, that when the model is correct (i.e., the model can express the learning target function), the standard MLE is consistent even under the covariate shift.

Table 1: Comparison of generalization error estimation methods under the covariate shift. $p_x(\boldsymbol{x})$ and $p_t(\boldsymbol{x})$ are probability density functions of training input points and test input points, respectively.

	MAIC [35]	SIC [43]	Proposed
Exactly unbiased when realizable	No	Yes	Yes
Asymptotically unbiased when unrealizable	Yes	No	Yes
$p_x(\boldsymbol{x})$ should be known	Yes	No	Yes
$p_t(\boldsymbol{x})$ should be known	Yes	Yes	Yes

of training examples. Therefore, it can have a large bias in small sample cases. The other attempt is a function approximation approach in terms of a *fixed* location of training input points [43], which yields an exactly unbiased estimator of the generalization error for finite samples under the realizability assumption (i.e., the learning target function is included in the model). The generalization error estimator is called the *subspace information criterion* (SIC). Although in the original literature, SIC is not explicitly derived for the covariate shift, it is applicable since the fixed training input points can be regarded as realizations of any distribution. Therefore, the unbiasedness of SIC is still maintained even when the training and test distributions are different. However, since this approach assumes the realizability of the learning target function, SIC can be inaccurate in unrealizable cases. Indeed, it is biased in unrealizable cases and the bias does not vanish even asymptotically.

In this paper, we try to integrate the advantages of the above methods. More specifically, we apply the density modification idea used in the former asymptotic approach to the latter function approximation approach. As a result, we obtain a generalization error estimator for the squared loss function which is exactly unbiased with finite samples in realizable cases and asymptotically unbiased in general (see Table 1).

Conventionally, the accuracy of generalization error estimators are investigated in terms of their unbiasedness [24, 1, 50, 20, 35, 43]. On the other hand, the purpose of estimating the generalization error is to discriminate good models from poor ones. To this end, we would like to accurately estimate the *difference* of the generalization error among different models. We show that, in addition to the unbiasedness, the proposed generalization error estimator can accurately estimate the difference of the generalization error for a general class of models.

The rest of this paper is organized as follows. In Section 2, the learning problem is formulated. In Section 3, a generalization error estimator is derived and its properties are investigated. In Section 4, numerical examples of regression for extrapolation and classification with imbalanced data are shown. Finally, in Section 5, conclusions and future prospects are described.

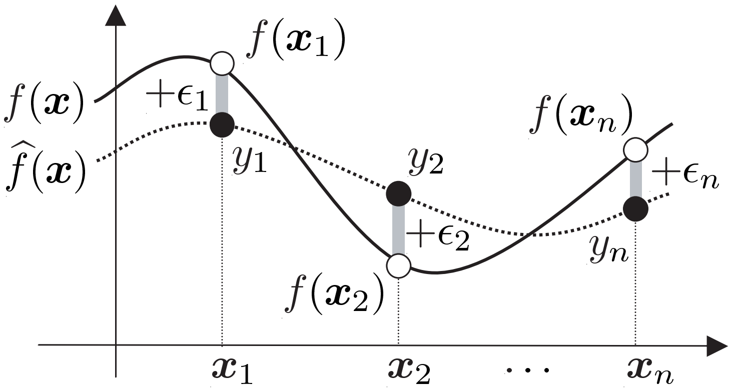


Figure 1: Supervised regression problem.

2 Problem Formulation

In this section, we formulate the learning problem discussed in this paper.

Let $f(\boldsymbol{x})$ be a fixed, real-valued function of d variables defined on the domain $\mathcal{D}$ ($\subset \mathbb{R}^d$), which is our *learning target function.* Since we deal with the values of $f(\boldsymbol{x})$ at some input points, we suppose that $f(\boldsymbol{x})$ is pointwise-defined[2]. We are given a set of n *training examples*, each of which consists of a *training input point* $\boldsymbol{x}_i$ in $\mathcal{D}$ and a *training output value* y_i in $\mathbb{R}$. The training input points $\{\boldsymbol{x}_i\}_{i=1}^n$ are drawn independently from a distribution with the probability density function $p_x(\boldsymbol{x})$. We suppose that $p_x(\boldsymbol{x})$ is strictly positive for any $\boldsymbol{x}$ in $\mathcal{D}$. The training output value y_i is degraded by unknown independent additive *noise* ϵ_i with mean zero and unknown variance σ^2 (Figure 1).

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

For the moment, we assume that $p_x(\boldsymbol{x})$ is known. In active learning scenarios, for example, $p_x(\boldsymbol{x})$ is naturally available since it is designed by users. Later, we theoretically and experimentally investigate the cases where $p_x(\boldsymbol{x})$ is unknown.

We use the following *linear regression model* for learning[3].

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

Here $\Phi = \{\varphi_i(\boldsymbol{x})\}_{i=1}^p$ are fixed linearly independent functions which are pointwise-defined, $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_p)^\top$ are parameters to be learned, and $^\top$ denotes the transpose of a vector or matrix. We assume that the number p of basis functions satisfies

$$p < n. \tag{3}$$

[2] Theoretically, we do not require $f(\boldsymbol{x})$ to be *smooth.* This implies that we are not trying to estimate the value of the function f at some input points, but we are trying to obtain an approximation $\widehat{f}(\boldsymbol{x})$ which minimizes the generalization error J defined by Eq.(13). However, practically, we may consider a smooth function $f(\boldsymbol{x})$.

[3] Note that, under some mild conditions, the results in this paper are still valid even in non-parametric cases where p is increased as $p = o(n^{\frac{1}{2}})$, which is discussed later.

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

$$\boldsymbol{X}_{i,j} = \varphi_j(\boldsymbol{x}_i). \tag{4}$$

We assume

$$\mathrm{rank}\,(\boldsymbol{X}) = p. \tag{5}$$

The parameters $\{\alpha_i\}_{i=1}^{p}$ in the regression model (2) are learned by a linear learning method, i.e., with an $p \times n$ matrix $\boldsymbol{L}$ which does not depend on the noise $\{\epsilon_i\}_{i=1}^{n}$, the learned parameter vector $\widehat{\boldsymbol{\alpha}} = (\widehat{\alpha}_1, \widehat{\alpha}_2, \ldots, \widehat{\alpha}_p)^\top$ is given by

$$\widehat{\boldsymbol{\alpha}} = \boldsymbol{L}\boldsymbol{y}, \tag{6}$$

where

$$\boldsymbol{y} = (y_1, y_2, \ldots, y_n)^\top. \tag{7}$$

The matrix $\boldsymbol{L}$ is called the *learning matrix*. We suppose that $\boldsymbol{L}$ satisfies for sufficiently large n

$$\boldsymbol{L} = \mathcal{O}_p(n^{-1}), \tag{8}$$

where the order notation for a matrix means that all the elements are of that order. Note that the above asymptotic order is in probability because $\boldsymbol{L}$ can depend on the random variables $\{\boldsymbol{x}_i\}_{i=1}^{n}$ (but it does not depend on $\{\epsilon_i\}_{i=1}^{n}$, as assumed above). Standard linear learning methods such as the weighted least-squares learning with a quadratic regularizer generally satisfy the condition (8):

$$\min_{\{\alpha_i\}_{i=1}^{p}} \left[\frac{1}{n} \sum_{i=1}^{n} w(\boldsymbol{x}_i) \left(\widehat{f}(\boldsymbol{x}_i) - y_i \right)^2 + \langle \boldsymbol{R}\boldsymbol{\alpha}, \boldsymbol{\alpha} \rangle \right], \tag{9}$$

where $w(\boldsymbol{x})$ is a strictly positive function called the *weight function* and $\boldsymbol{R}$ is a p-dimensional positive semi-definite matrix called the *regularization matrix*.

The aim of learning is to obtain a function $\widehat{f}(\boldsymbol{x})$ which attains a small *generalization error*. In this paper, the generalization error is measured by the expected squared error over all test input points. We suppose that the test input points are drawn independently from a distribution with the probability density function $p_t(\boldsymbol{x})$, which is assumed to be strictly positive for any $\boldsymbol{x}$ in $\mathcal{D}$. For the moment, we treat $p_t(\boldsymbol{x})$ as a known function. Later, we theoretically and experimentally investigate the cases where $p_t(\boldsymbol{x})$ is unknown[4].

Let us consider a functional Hilbert space $\mathcal{H}$ spanned by the following functions.

$$\{f \mid \|f\|_{\mathcal{H}} < \infty\}, \tag{10}$$

where the inner product and norm are defined by

$$\begin{aligned} \langle f, g \rangle_{\mathcal{H}} &= \int_{\mathcal{D}} f(\boldsymbol{x}) g(\boldsymbol{x}) p_t(\boldsymbol{x}) d\boldsymbol{x}, & (11) \\ \|f\|_{\mathcal{H}} &= \sqrt{\langle f, f \rangle_{\mathcal{H}}}. & (12) \end{aligned}$$

[4] In interpolation or extrapolation scenarios, we may *define* $p_t(\boldsymbol{x})$ by ourselves because it can be interpreted as a weight function representing the degree-of-interestingness of the region.

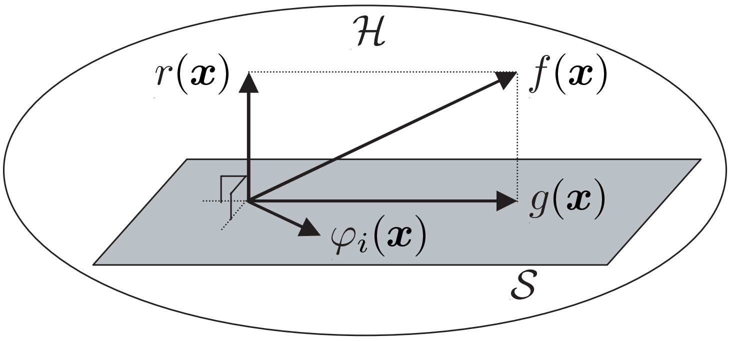


Figure 2: Decomposition of $f(\boldsymbol{x})$. $\mathcal{S}$ is the subspace spanned by $\{\varphi_i(\boldsymbol{x})\}_{i=1}^p$.

We suppose that the learning target function $f(\boldsymbol{x})$ and the basis functions $\{\varphi_i(\boldsymbol{x})\}_{i=1}^p$ are included in the above function space $\mathcal{H}$. Then the generalization error is expressed as

$$\begin{aligned} J &= \int_{\mathcal{D}} \left(\widehat{f}(\boldsymbol{x}) - f(\boldsymbol{x})\right)^2 p_t(\boldsymbol{x}) d\boldsymbol{x} \\ &= \|\widehat{f} - f\|_{\mathcal{H}}^2. \end{aligned} \tag{13}$$

Note that $J = 0$ does not generally imply $\widehat{f}(\boldsymbol{x}) = f(\boldsymbol{x})$ for *all* $\boldsymbol{x}$ in $\mathcal{D}$. If the functional Hilbert space $\mathcal{H}$ has the reproducing kernel [3, 30, 50, 31, 49, 32], $J = 0$ if and only if $\widehat{f}(\boldsymbol{x}) = f(\boldsymbol{x})$ for all $\boldsymbol{x}$ in $\mathcal{D}$.

A main purpose of this paper is to give an estimator $\widehat{J}$ of the generalization error J which is useful for comparing the generalization error among different *models*. Here, a model refers to the basis functions Φ and some factors which control the learning matrix $\boldsymbol{L}$ (e.g., $w(\boldsymbol{x})$ or $\boldsymbol{R}$ in Eq.(9)).

3 Generalization Error Estimator

In this section, we derive an estimator of the generalization error J, investigate its theoretical properties, and discuss its relation to existing methods.

3.1 Derivation

Since the learning target function $f(\boldsymbol{x})$ belongs to the Hilbert space $\mathcal{H}$, its projection onto any subspace always exists. Therefore, without loss of generality, it can be decomposed as (see Figure 2)

$$f(\boldsymbol{x}) = g(\boldsymbol{x}) + r(\boldsymbol{x}), \tag{14}$$

where the first component is the orthogonal projection of $f(\boldsymbol{x})$ onto the span of $\{\varphi_i(\boldsymbol{x})\}_{i=1}^p$ and the second component is orthogonal to $\{\varphi_i(\boldsymbol{x})\}_{i=1}^p$, i.e., for $i = 1, 2, \ldots, p$,

$$\langle r, \varphi_i \rangle_{\mathcal{H}} = \int_{\mathcal{D}} r(\boldsymbol{x})\varphi_i(\boldsymbol{x})p_t(\boldsymbol{x})d\boldsymbol{x} = 0. \tag{15}$$

Since $g(\boldsymbol{x})$ is included in the span of $\{\varphi_i(\boldsymbol{x})\}_{i=1}^{p}$, it is expressed by

$$g(\boldsymbol{x}) = \sum_{i=1}^{p} \alpha_i^* \varphi_i(\boldsymbol{x}), \tag{16}$$

where $\boldsymbol{\alpha}^* = (\alpha_1^*, \alpha_2^*, \ldots, \alpha_p^*)^\top$ are unknown optimal parameters. Note that $r(\boldsymbol{x})$ is pointwise-defined because both $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$ are pointwise-defined. In the following, we say that $f(\boldsymbol{x})$ is *realizable* if $r(\boldsymbol{x}) = 0$ for all $\boldsymbol{x}$ in $\mathcal{D}$.

Let $\boldsymbol{U}$ be a p-dimensional matrix with the (i,j)-th element

$$\boldsymbol{U}_{i,j} = \langle \varphi_i, \varphi_j \rangle_{\mathcal{H}} = \int_{\mathcal{D}} \varphi_i(\boldsymbol{x}) \varphi_j(\boldsymbol{x}) p_t(\boldsymbol{x}) d\boldsymbol{x}, \tag{17}$$

which is assumed to be accessible in the current setting. Then the generalization error J is expressed as

$$\begin{aligned} J &= \|\widehat{f}\|_{\mathcal{H}}^2 - 2\langle \widehat{f}, g + r \rangle_{\mathcal{H}} + \|f\|_{\mathcal{H}}^2 \\ &= \langle \boldsymbol{U}\widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\alpha}} \rangle - 2\langle \boldsymbol{U}\widehat{\boldsymbol{\alpha}}, \boldsymbol{\alpha}^* \rangle + C, \end{aligned} \tag{18}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product in $\mathbb{R}^p$ and

$$C = \|f\|_{\mathcal{H}}^2 = \int_{\mathcal{D}} f(\boldsymbol{x})^2 p_t(\boldsymbol{x}) d\boldsymbol{x}. \tag{19}$$

In Eq.(18), the first term $\langle \boldsymbol{U}\widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\alpha}} \rangle$ is accessible and the third term C is a constant (i.e., it does not depend on the model). Therefore, we focus on estimating the second term '$-2\langle \boldsymbol{U}\widehat{\boldsymbol{\alpha}}, \boldsymbol{\alpha}^* \rangle$'.

Hypothetically, let us suppose that a learning matrix $\boldsymbol{L}_u$ which gives a linear unbiased estimator of the unknown true parameter $\boldsymbol{\alpha}^*$ is available:

$$\mathbb{E}_{\boldsymbol{\epsilon}} \boldsymbol{L}_u \boldsymbol{y} = \boldsymbol{\alpha}^*, \tag{20}$$

where $\mathbb{E}_{\boldsymbol{\epsilon}}$ denotes the expectation over the noise $\{\epsilon_i\}_{i=1}^{n}$. Note that $\boldsymbol{L}_u$ does not depend on $\boldsymbol{L}$. Then it holds that

$$\begin{aligned} \mathbb{E}_{\boldsymbol{\epsilon}} \langle \boldsymbol{U}\widehat{\boldsymbol{\alpha}}, \boldsymbol{\alpha}^* \rangle &= \langle \mathbb{E}_{\boldsymbol{\epsilon}} \boldsymbol{U}\boldsymbol{L}\boldsymbol{y}, \mathbb{E}_{\boldsymbol{\epsilon}} \boldsymbol{L}_u \boldsymbol{y} \rangle \\ &= \mathbb{E}_{\boldsymbol{\epsilon}} \langle \boldsymbol{U}\boldsymbol{L}\boldsymbol{y}, \boldsymbol{L}_u \boldsymbol{y} \rangle - \sigma^2 \mathrm{tr}(\boldsymbol{U}\boldsymbol{L}\boldsymbol{L}_u^\top). \end{aligned} \tag{21}$$

If an unbiased estimator σ_u^2 of the noise variance σ^2 is available, an unbiased estimator of $\mathbb{E}_{\boldsymbol{\epsilon}} \langle \boldsymbol{U}\widehat{\boldsymbol{\alpha}}, \boldsymbol{\alpha}^* \rangle$ can be obtained by $\langle \boldsymbol{U}\boldsymbol{L}\boldsymbol{y}, \boldsymbol{L}_u \boldsymbol{y} \rangle - \sigma_u^2 \mathrm{tr}(\boldsymbol{U}\boldsymbol{L}\boldsymbol{L}_u^\top)$:

$$\mathbb{E}_{\boldsymbol{\epsilon}} [\langle \boldsymbol{U}\boldsymbol{L}\boldsymbol{y}, \boldsymbol{L}_u \boldsymbol{y} \rangle - \sigma_u^2 \mathrm{tr}(\boldsymbol{U}\boldsymbol{L}\boldsymbol{L}_u^\top)] = \mathbb{E}_{\boldsymbol{\epsilon}} \langle \boldsymbol{U}\widehat{\boldsymbol{\alpha}}, \boldsymbol{\alpha}^* \rangle. \tag{22}$$

However, either $\boldsymbol{L}_u$ or σ_u^2 could be unavailable[5]. So we use the following approximations instead:

$$\widehat{\boldsymbol{L}}_u = (\boldsymbol{X}^\top \boldsymbol{D}\boldsymbol{X})^{-1}\boldsymbol{X}^\top \boldsymbol{D}, \tag{23}$$

$$\widehat{\sigma_u^2} = \frac{\|\boldsymbol{G}\boldsymbol{y}\|^2}{\mathrm{tr}(\boldsymbol{G})}, \tag{24}$$

where $\boldsymbol{D}$ is the diagonal matrix with the i-th diagonal element

$$\boldsymbol{D}_{i,i} = \frac{p_t(\boldsymbol{x}_i)}{p_x(\boldsymbol{x}_i)}, \tag{25}$$

and

$$\boldsymbol{G} = \boldsymbol{I} - \boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{X})^{-1}\boldsymbol{X}^\top. \tag{26}$$

Note that $\boldsymbol{I}$ denotes the identity matrix.

Eq.(23) is the learning matrix which corresponds to the following weighted least-squares learning.

$$\min_{\{\alpha_i\}_{i=1}^p} \left[\frac{1}{n}\sum_{i=1}^n \frac{p_t(\boldsymbol{x}_i)}{p_x(\boldsymbol{x}_i)}\Big(\sum_{j=1}^p \alpha_j\varphi_j(\boldsymbol{x}_i) - y_i\Big)^2\right]. \tag{27}$$

It can be proved that the above $\widehat{\boldsymbol{L}}_u$ exactly fulfills Eq.(20) in realizable cases and it asymptotically satisfies Eq.(20) in general (see [35] and Lemma 1 shown later).

On the other hand, it is well known that $\widehat{\sigma_u^2}$ is an exact unbiased estimator of σ^2 in realizable cases [9]. In general cases, however, it is not unbiased even asymptotically. Although it is possible to obtain asymptotic unbiased estimators of σ^2 under some smoothness assumption on $f(\boldsymbol{x})$ [37], we do not use such asymptotic unbiased estimators because it turns out shortly that the asymptotic unbiasedness of $\widehat{\sigma_u^2}$ is not important in the following.

Based on the above discussion, we define the following estimator $\widehat{J}$ of the generalization error J.

$$\widehat{J} = \langle \boldsymbol{U}\boldsymbol{L}\boldsymbol{y}, \boldsymbol{L}\boldsymbol{y}\rangle - 2\langle \boldsymbol{U}\boldsymbol{L}\boldsymbol{y}, \widehat{\boldsymbol{L}}_u\boldsymbol{y}\rangle + 2\widehat{\sigma_u^2}\mathrm{tr}(\boldsymbol{U}\boldsymbol{L}\widehat{\boldsymbol{L}}_u^\top), \tag{28}$$

where the first term is $\langle \boldsymbol{U}\widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\alpha}}\rangle$ and the second and third terms correspond to $-2\langle \boldsymbol{U}\widehat{\boldsymbol{\alpha}}, \boldsymbol{\alpha}^*\rangle$ (cf. Eq.(18)).

3.2 Unbiasedness

Here we investigate the unbiasedness of $\widehat{J}$.

Let $B_{\boldsymbol{\epsilon}}$ be the bias of $\widehat{J}$ with respect to the noise $\{\epsilon_i\}_{i=1}^n$:

$$B_{\boldsymbol{\epsilon}} = \mathbb{E}_{\boldsymbol{\epsilon}}[\widehat{J} - J] + C, \tag{29}$$

[5] Note that $\boldsymbol{L}_u$ is always available if the functional Hilbert space $\mathcal{H}$ has the reproducing kernel and the span of the basis functions $\{\varphi_i(\boldsymbol{x})\}_{i=1}^p$ is included in the span of $\{K(\boldsymbol{x}, \boldsymbol{x}_i)\}_{i=1}^n$ [40], where $K(\boldsymbol{x}, \boldsymbol{x}')$ is the reproducing kernel. In this paper, however, we consider general functional Hilbert spaces and general basis functions which may not satisfy such conditions.

where the constant C (see Eq.(19)) is added for making the following discussion simple. Then we have the following lemma (proofs of all lemmas are provided in Appendix).

Lemma 1 *If* $r(\boldsymbol{x}_i) = 0$ *for* $i = 1, 2, \ldots, n$,

$$B_{\boldsymbol{\epsilon}} = 0. \tag{30}$$

If $\delta = \max\{|r(\boldsymbol{x}_i)|\}_{i=1}^n$ *is sufficiently small,*

$$B_{\boldsymbol{\epsilon}} = \mathcal{O}(\delta). \tag{31}$$

If n *is sufficiently large,*

$$B_{\boldsymbol{\epsilon}} = \mathcal{O}_p(n^{-\frac{1}{2}}). \tag{32}$$

Note that in Eq.(32), the asymptotic order is in probability because the expectation over $\{\boldsymbol{x}_i\}_{i=1}^n$ is not taken. The above lemma implies that, except for the constant C, $\widehat{J}$ is exactly unbiased if $f(\boldsymbol{x})$ is strictly realizable (see Eq.(30)), it is almost unbiased if $f(\boldsymbol{x})$ is almost realizable (see Eq.(31)), and it is asymptotically unbiased in general (see Eq.(32)). Note that $\widehat{J}$ is still asymptotically unbiased (i.e., $B_{\boldsymbol{\epsilon}} = o_p(1)$) even in non-parametric cases if p is increased as $p = o(n^{\frac{1}{2}})$ (see Remark in Appendix A for detail).

3.3 Effectiveness in Model Comparison

A purpose of estimating the generalization error is model selection, i.e., to distinguish good models from poor ones. To this end, the *difference* of the generalization error among different models should be accurately estimated. Here, we show that the proposed generalization error estimator $\widehat{J}$ is useful for this purpose. Recall that our model has basis functions Φ and some factors which control the learning matrix $\boldsymbol{L}$ (e.g., the weight function or the regularization matrix, see Section 2).

Let ΔJ, $\Delta\widehat{J}$, and $\Delta B_{\boldsymbol{\epsilon}}$ be the differences of J, $\widehat{J}$, and $B_{\boldsymbol{\epsilon}}$ for two models, respectively:

$$\Delta B_{\boldsymbol{\epsilon}} = \mathbb{E}_{\boldsymbol{\epsilon}}[\Delta\widehat{J} - \Delta J]. \tag{33}$$

If the "size" of $\Delta B_{\boldsymbol{\epsilon}}$ is smaller than that of $\mathbb{E}_{\boldsymbol{\epsilon}}[\Delta J]$, then $\widehat{J}$ is expected to be useful for comparing the generalization error among different models. Let $\mathcal{M}$ be a set of models. We say that a generalization error estimator $\widehat{J}$ is *effective in model comparison for* $\mathcal{M}$ if

$$|\Delta B_{\boldsymbol{\epsilon}}| < |\mathbb{E}_{\boldsymbol{\epsilon}}[\Delta J]| \tag{34}$$

for any two different models in $\mathcal{M}$. Also, we say that $\widehat{J}$ is *asymptotically effective in model comparison for* $\mathcal{M}$ if any two different models in $\mathcal{M}$ satisfy[6]

$$\Delta B_{\boldsymbol{\epsilon}} = \mathcal{O}_p(n^{-s}) \quad \text{and} \quad \mathbb{E}_{\boldsymbol{\epsilon}}[\Delta J] = \mathcal{O}_p(n^{-t}) \quad \text{with } s > t. \tag{35}$$

In the following, we investigate the (asymptotic) effectiveness of $\widehat{J}$ in model comparison.

First, we have the following corollary immediately from Eq.(32).

[6] This definition of asymptotic effectiveness claims that the asymptotic *upper bound* on $\Delta B_{\boldsymbol{\epsilon}}$ is smaller than that of $\mathbb{E}_{\boldsymbol{\epsilon}}[\Delta J]$. Another possible definition would be $\Delta B_{\boldsymbol{\epsilon}} = o_p(\mathbb{E}_{\boldsymbol{\epsilon}}[\Delta J])$, which remains to be investigated.

Corollary 2 *If two learned functions obtained from two different models converge to different functions,*

$$\Delta B_{\boldsymbol{\epsilon}} = \mathcal{O}_p(n^{-\frac{1}{2}}) \qquad \textit{and} \qquad \mathbb{E}_{\boldsymbol{\epsilon}}[\Delta J] = \mathcal{O}_p(1). \tag{36}$$

For models with totally different Φ (i.e., the intersection of the spans of the basis functions is zero), learned functions obtained from such models generally converge to different functions. Therefore, in comparison of such models, $\widehat{J}$ is asymptotically effective.

In the following, we investigate the models such that Φ is common but other factors which control the learning matrix $\boldsymbol{L}$ are different. Let us denote the set of such models by $\mathcal{M}_\Phi$, indicating that Φ is common. Then, from Eq.(30), we immediately have the following corollary.

Corollary 3 *If $r(\boldsymbol{x}_i) = 0$ for $i = 1, 2, \ldots, n$, then for any two models in $\mathcal{M}_\Phi$,*

$$\Delta B_{\boldsymbol{\epsilon}} = 0. \tag{37}$$

This implies that if $f(\boldsymbol{x})$ is realizable and $|\mathbb{E}_{\boldsymbol{\epsilon}}[\Delta J]| > 0$, $\widehat{J}$ is effective in model comparison for $\mathcal{M}_\Phi$. Similarly, Eq.(31) yields that

$$\Delta B_{\boldsymbol{\epsilon}} = \mathcal{O}(\delta) \tag{38}$$

for any two models in $\mathcal{M}_\Phi$. Therefore, if $f(\boldsymbol{x})$ is almost realizable, $\widehat{J}$ would be useful for model comparison.

Finally, we consider the case where Φ is common but the learning target function is unrealizable. Let $\boldsymbol{L}_1$ and $\boldsymbol{L}_2$ be the learning matrices obtained from two different models, and let

$$\Delta \boldsymbol{L} = \boldsymbol{L}_2 - \boldsymbol{L}_1. \tag{39}$$

Then we have the following lemma.

Lemma 4 *Suppose $(\boldsymbol{L}_1 + \boldsymbol{L}_2)\boldsymbol{z}_f - 2\boldsymbol{\alpha}^* = \mathcal{O}_p(n^{-u})$ with $u < \frac{1}{2}$. Then, for*

$$\Delta \boldsymbol{L} = \mathcal{O}_p(n^{-t}), \tag{40}$$

we have

$$\Delta B_{\boldsymbol{\epsilon}} = \mathcal{O}_p(n^{-(t-\frac{1}{2})}) \qquad \textit{and} \qquad \mathbb{E}_{\boldsymbol{\epsilon}}[\Delta J] = \mathcal{O}_p(n^{-(t-1+u)}) \tag{41}$$

This lemma implies that under some condition, $\widehat{J}$ is asymptotically effective in model comparison for $\mathcal{M}_\Phi$. Note that even in non-parametric cases where p increases as n increases, $\widehat{J}$ is still asymptotically effective in model comparison for $\mathcal{M}_\Phi$ (see Remark in Appendix B for detail).

3.4 Relation to Other Methods

Estimating the generalization error when training and test input points are drawn from different distributions has already been studied by modifying AIC [35] which is explicitly derived for this situation, and by defining the subspace information criterion (SIC) [43] which is not explicitly proposed for this situation but is applicable. Here we relate our method with these methods (see also Table 1).

The modified AIC (MAIC) is an asymptotic unbiased estimator of the generalization error for statistically regular models with the maximum weighted log-likelihood estimation. For linear regression models with independent and identically distributed Gaussian noise, the maximum weighted log-likelihood estimation is reduced to the weighted least-squares learning (see e.g., Eqs.(56) and (57)), and MAIC is expressed after some shift and rescale as follows.

$$\widehat{J}_{MAIC} = \langle \widehat{\boldsymbol{U}}\boldsymbol{L}\boldsymbol{y}, \boldsymbol{L}\boldsymbol{y}\rangle - 2\langle \widehat{\boldsymbol{U}}\boldsymbol{L}\boldsymbol{y}, \widehat{\boldsymbol{L}}_u\boldsymbol{y}\rangle + 2\mathrm{tr}(\widehat{\boldsymbol{U}}\boldsymbol{L}\widehat{\boldsymbol{C}}\widehat{\boldsymbol{L}}_u^\top), \tag{42}$$

where

$$\widehat{\boldsymbol{U}} = \tfrac{1}{n}\boldsymbol{X}^\top \boldsymbol{D}\boldsymbol{X}, \tag{43}$$

and $\widehat{\boldsymbol{C}}$ is the diagonal matrix with the i-th diagonal element

$$\widehat{\boldsymbol{C}}_{i,i} = (y_i - \widehat{f}(\boldsymbol{x}_i))^2. \tag{44}$$

Note MAIC also assumes that both $p_x(\boldsymbol{x})$ and $p_t(\boldsymbol{x})$ are known (see Table 1).

The appearances of $\widehat{J}_{MAIC}$ and $\widehat{J}$ are similar but different in two aspects (cf. Eq.(28)).

(i) The matrix $\boldsymbol{U}$ in $\widehat{J}$ is replaced by its empirical estimate $\widehat{\boldsymbol{U}}$ in $\widehat{J}_{MAIC}$.

(ii) Instead of $\widehat{\boldsymbol{C}}$ in $\widehat{J}_{MAIC}$, $\widehat{\sigma_u^2}\boldsymbol{I}$ is used in $\widehat{J}$.

The former difference is especially interesting because $\widehat{J}_{MAIC}$ does not use the true matrix $\boldsymbol{U}$, although it is accessible by the assumption. In Section 4, we experimentally show that using $\widehat{\boldsymbol{U}}$ can cause an unstable behavior when the dimension d of the input vector $\boldsymbol{x}$ is high.

The above $\widehat{J}_{MAIC}$ satisfies

$$\mathbb{E}_{\boldsymbol{x}}\mathbb{E}_{\boldsymbol{\epsilon}}[\widehat{J}_{MAIC} - J] = o(n^{-1}) - C, \tag{45}$$

where $\mathbb{E}_{\boldsymbol{x}}$ denotes the expectation over training input points $\{\boldsymbol{x}_i\}_{i=1}^n$. This shows that $\widehat{J}_{MAIC}$ has a fast asymptotic convergence with respect to training input points and noise. On the other hand, if $\mathbb{E}_{\boldsymbol{x}}$ is not taken, $\widehat{J}_{MAIC}$ satisfies

$$\mathbb{E}_{\boldsymbol{\epsilon}}[\widehat{J}_{MAIC} - J] = \mathcal{O}_p(n^{-\frac{1}{2}}) - C, \tag{46}$$

which means that $\widehat{J}_{MAIC}$ has the same asymptotic order as $\widehat{J}$ (see Eq.(32)). However, a crucial difference is that $\widehat{J}_{MAIC}$ does *not* satisfy Eqs.(30) and (31) even when (almost) realizability is satisfied. It seems that the effectiveness of $\widehat{J}_{MAIC}$ in model comparison has not been explicitly investigated so far [47, 26], although the difference of AIC has been investigated thoroughly [23, 33, 34].

Another related method, SIC, is an estimator of the squared distance between the learned and learning target functions in a function space. It was shown that in realizable cases, SIC is exactly unbiased for any fixed locations of training input points. Under the setting in the current paper, the above squared distance corresponds to J and SIC is expressed as

$$\widehat{J}_{SIC} = \langle \boldsymbol{U}\boldsymbol{L}\boldsymbol{y}, \boldsymbol{L}\boldsymbol{y}\rangle - 2\langle \boldsymbol{U}\boldsymbol{L}\boldsymbol{y}, \widetilde{\boldsymbol{L}}_u\boldsymbol{y}\rangle + 2\widehat{\sigma_u^2}\mathrm{tr}(\boldsymbol{U}\boldsymbol{L}\widetilde{\boldsymbol{L}}_u^\top), \tag{47}$$

where

$$\widetilde{\boldsymbol{L}}_u = (\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top. \tag{48}$$

Note that in $\widehat{J}_{SIC}$, $p_t(\boldsymbol{x})$ is assumed to be known but $p_x(\boldsymbol{x})$ is not needed. The appearances of $\widehat{J}_{SIC}$ and $\widehat{J}$ are rather similar, but $\widehat{\boldsymbol{L}}_u$ in $\widehat{J}$ is replaced by $\widetilde{\boldsymbol{L}}_u$ in $\widehat{J}_{SIC}$ (cf. Eq.(28)).

The fixed training input points can be regarded as realizations of any distribution. Therefore, SIC is still applicable to the cases where the distributions of training and test input points are different. Indeed, we can prove that $\widehat{J}_{SIC}$ satisfies Eqs.(30) and (31) even when the training and test distributions are different, although this fact was not explicitly pointed out in the original paper. However, we can also prove that $\widehat{J}_{SIC}$ is *not* asymptotically unbiased in unrealizable cases. This difference is very significant as experimentally shown in Section 4. Note that $\widehat{J}_{SIC}$ satisfies Eq.(32) if the training and test distributions are the same [44].

3.5 When $p_x(\boldsymbol{x})$ and $p_t(\boldsymbol{x})$ Are Unknown

So far, we assumed that both $p_x(\boldsymbol{x})$ and $p_t(\boldsymbol{x})$ are known. Here we consider the cases where they are unknown.

$p_t(\boldsymbol{x})$ is contained in $\boldsymbol{U}$ and $\widehat{\boldsymbol{L}}_u$, while $p_x(\boldsymbol{x})$ appears only in $\widehat{\boldsymbol{L}}_u$. So we investigate the effect of replacing $p_x(\boldsymbol{x})$ and $p_t(\boldsymbol{x})$ included in $\boldsymbol{U}$ and $\widehat{\boldsymbol{L}}_u$ with their estimates[7].

First, we consider the case where $p_t(\boldsymbol{x})$ is unknown but its approximation $\widehat{p}_t(\boldsymbol{x})$ is available. Let $\widehat{J}_t$ be $\widehat{J}$ calculated with $\widehat{p}_t(\boldsymbol{x})$ instead of $p_t(\boldsymbol{x})$. Then we have the following lemma.

Lemma 5 *Let*

$$\begin{aligned} \eta_t &= \max\{|\widehat{p}_t(\boldsymbol{x}_i) - p_t(\boldsymbol{x}_i)|\}_{i=1}^n, & (49)\\ \xi_t &= \max\left\{\left|\int_{\mathcal{D}} \varphi_i(\boldsymbol{x})\varphi_j(\boldsymbol{x})\left(\widehat{p}_t(\boldsymbol{x}) - p_t(\boldsymbol{x})\right)d\boldsymbol{x}\right|\right\}_{i,j=1}^n. & (50)\end{aligned}$$

[7] Note that $p_x(\boldsymbol{x})$ and $p_t(\boldsymbol{x})$ can also appear in the learning matrix $\boldsymbol{L}$ (e.g., Eq.(57)). In such a case, a learning matrix obtained using some estimates of $p_x(\boldsymbol{x})$ and $p_t(\boldsymbol{x})$ is generally different from the learning matrix obtained using the true densities $p_x(\boldsymbol{x})$ and $p_t(\boldsymbol{x})$. However, we here aim to investigate the accuracy of $\widehat{J}$ as a function of $\boldsymbol{L}$, so whether $\boldsymbol{L}$ includes $p_x(\boldsymbol{x})$ and $p_t(\boldsymbol{x})$ does not matter.

If η_t and ξ_t are sufficiently small,

$$\widehat{J}_t = \widehat{J} + \mathcal{O}(\eta_t + \xi_t). \tag{51}$$

This lemma states that if a reasonably good estimator $\widehat{p}_t(\boldsymbol{x})$ of the true density function $p_t(\boldsymbol{x})$ is available, a good approximation of $\widehat{J}$ can be obtained. Suppose we are given a large number of *unlabeled samples*, which are input points without output values independently drawn from the distribution with the probability density function $p_t(\boldsymbol{x})$. Actually, in some application domains—e.g., document classification [27] or bioinformatics [18]—a large number of unlabeled samples are easily gathered. In such cases, a reasonably good estimator $\widehat{p}_t(\boldsymbol{x})$ may be obtained by some standard density estimation methods.

Next, we consider the case where $p_x(\boldsymbol{x})$ is unknown but its approximation $\widehat{p}_x(\boldsymbol{x})$ is available. Let $\widehat{J}_x$ be $\widehat{J}$ calculated with $\widehat{p}_x(\boldsymbol{x})$ instead of $p_x(\boldsymbol{x})$. Since $p_x(\boldsymbol{x})$ is included in the denominator (see Eq.(25)), $\widehat{J}_x$ can be very different from $\widehat{J}$ even if $\widehat{p}_x(\boldsymbol{x})$ is a good estimator of $p_x(\boldsymbol{x})$. However, if mild assumptions on $p_x(\boldsymbol{x}_i)$ and $\widehat{p}_x(\boldsymbol{x}_i)$ are satisfied, we can guarantee the accuracy of $\widehat{J}_x$ as follows.

Lemma 6 *Let*

$$\begin{aligned} \eta_x &= \max\{|\widehat{p}_x(\boldsymbol{x}_i) - p_x(\boldsymbol{x}_i)|\}_{i=1}^n, && (52)\\ \gamma &= \min\{p_x(\boldsymbol{x}_i)\}_{i=1}^n, && (53)\\ \widehat{\gamma} &= \min\{\widehat{p}_x(\boldsymbol{x}_i)\}_{i=1}^n. && (54) \end{aligned}$$

If $\gamma > 0$ and $\widehat{\gamma} > 0$, and if η_x is sufficiently small,

$$\widehat{J}_x = \widehat{J} + \mathcal{O}\left(\frac{\eta_x}{\gamma\widehat{\gamma}}\right). \tag{55}$$

This lemma states that if $p_x(\boldsymbol{x}_i)$ and $\widehat{p}_x(\boldsymbol{x}_i)$ are lower bounded by some (not very small) positive constants and reasonably accurate estimates of the density values at the training input points $\{\boldsymbol{x}_i\}_{i=1}^n$ are available, a good approximation of $\widehat{J}$ can be obtained.

In practical situations with rather small training samples, accurately estimating the training input density $p_x(\boldsymbol{x})$ is difficult. However, the above lemma guarantees that as long as $\{p_x(\boldsymbol{x}_i)\}_{i=1}^n$, the density values at the training input points $\{\boldsymbol{x}_i\}_{i=1}^n$, can be estimated reasonably, a good approximation of $\widehat{J}$ would be obtained.

4 Numerical Examples

In this section, we show some numerical examples.

In all the simulations, we use the following weighted least-squares learning suggested in the reference [35].

$$\min_{\{\alpha_i\}_{i=1}^p} \left[\sum_{i=1}^n \left(\frac{p_t(\boldsymbol{x}_i)}{p_x(\boldsymbol{x}_i)}\right)^\lambda \left(\widehat{f}(\boldsymbol{x}_i) - y_i\right)^2\right], \tag{56}$$

where λ ($0 \leq \lambda \leq 1$) is a tuning parameter. The learning matrix of the above weighted least-squares learning is given by

$$\boldsymbol{L} = (\boldsymbol{X}^\top \boldsymbol{D}^\lambda \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{D}^\lambda. \tag{57}$$

Roughly speaking, $\lambda = 1$ (consistent weighted least-squares learning) has small bias but has large variance, while $\lambda = 0$ (ordinary least-squares learning) has comparatively small variance but has large bias. Therefore, changing λ between 0 and 1 would correspond to controlling the bias-variance trade-off. When the number n of training examples is large, a large λ which provides a small bias would be appropriate because the variance is relatively small. On the other hand, when the number n of training examples is small, the variance generally dominates the bias so a small λ which provides a comparatively small variance would be appropriate.

4.1 One-Dimensional Regression for Extrapolation

We first examine the behavior of the proposed generalization error estimator and other methods using a simple one-dimensional regression dataset. Let the learning target function $f(x)$ be the *sinc* function:

$$f(x) = \mathrm{sinc}(x). \tag{58}$$

Let $N(\mu, c^2)$ denote the normal distribution with mean μ and variance c^2, and let $\phi_{\mu,c^2}(x)$ be the probability density function of $N(\mu, c^2)$. Let the training and test input densities be

$$\begin{aligned} p_x(x) &= \phi_{1,(1/2)^2}(x), & (59) \\ p_t(x) &= \phi_{2,(1/4)^2}(x). & (60) \end{aligned}$$

This setting implies that we are considering an extrapolation problem (see Figure 3). For the moment, we suppose that both the training and test input densities are known. Later, we investigate the cases where the densities are unknown. Random noise $\{\epsilon_i\}_{i=1}^n$ are drawn independently from $N(0, \sigma^2)$ where $\sigma^2 = (1/4)^2$. σ^2 is treated as an unknown variable. We use a polynomial model of order $p-1$ for learning:

$$\varphi_i(x) = x^{i-1} \text{ for } i = 1, 2, \ldots, p. \tag{61}$$

Let us consider the following three cases:

$$(p, n) = (2, 150), (3, 100), (2, 15). \tag{62}$$

When $p = 2$, $\widehat{f}(\boldsymbol{x})$ is a linear function so $f(\boldsymbol{x})$ is heavily unrealizable (see Figure 3). On the other hand, when $p = 3$, $f(\boldsymbol{x})$ is rather close to realizable. Therefore, the above three cases roughly correspond to "*unrealizable and large samples*", "*realizable and small samples*", and "*unrealizable and small samples*". We randomly create $\{x_i, \epsilon_i\}_{i=1}^n$ and calculate the values of J, $\widehat{J}$, $\widehat{J}_{MAIC}$, $\widehat{J}_{SIC}$, and the 10-fold cross-validation (10CV) score

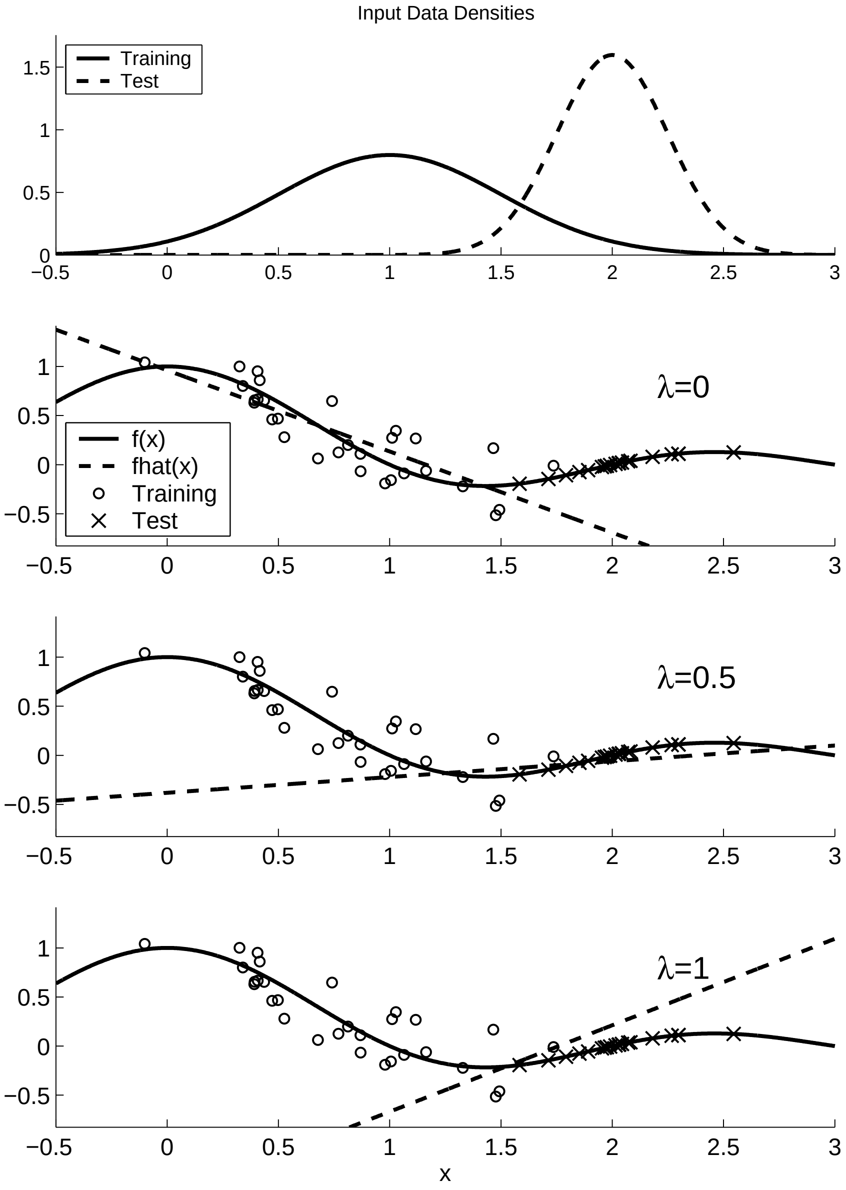


Figure 3: An illustrative example of extrapolation by fitting a linear function. The top graph depicts the probability density functions of the training and test input points. In the bottom three graphs, the learning target function $f(x)$ is drawn by the solid line, the noisy training examples are plotted with $\circ$'s, a learned function $\widehat{f}(x)$ is drawn by the dashed line, and the (noiseless) test examples are plotted with $\times$'s. Three different learned functions are obtained by weighted least-squares learning with different tuning parameter λ. $\lambda = 0$ corresponds to the ordinary least-squares learning, while $\lambda = 1$ corresponds to the consistent weighted least-squares learning. With finite samples, an intermediate λ, say $\lambda = 0.5$, often provides better results.

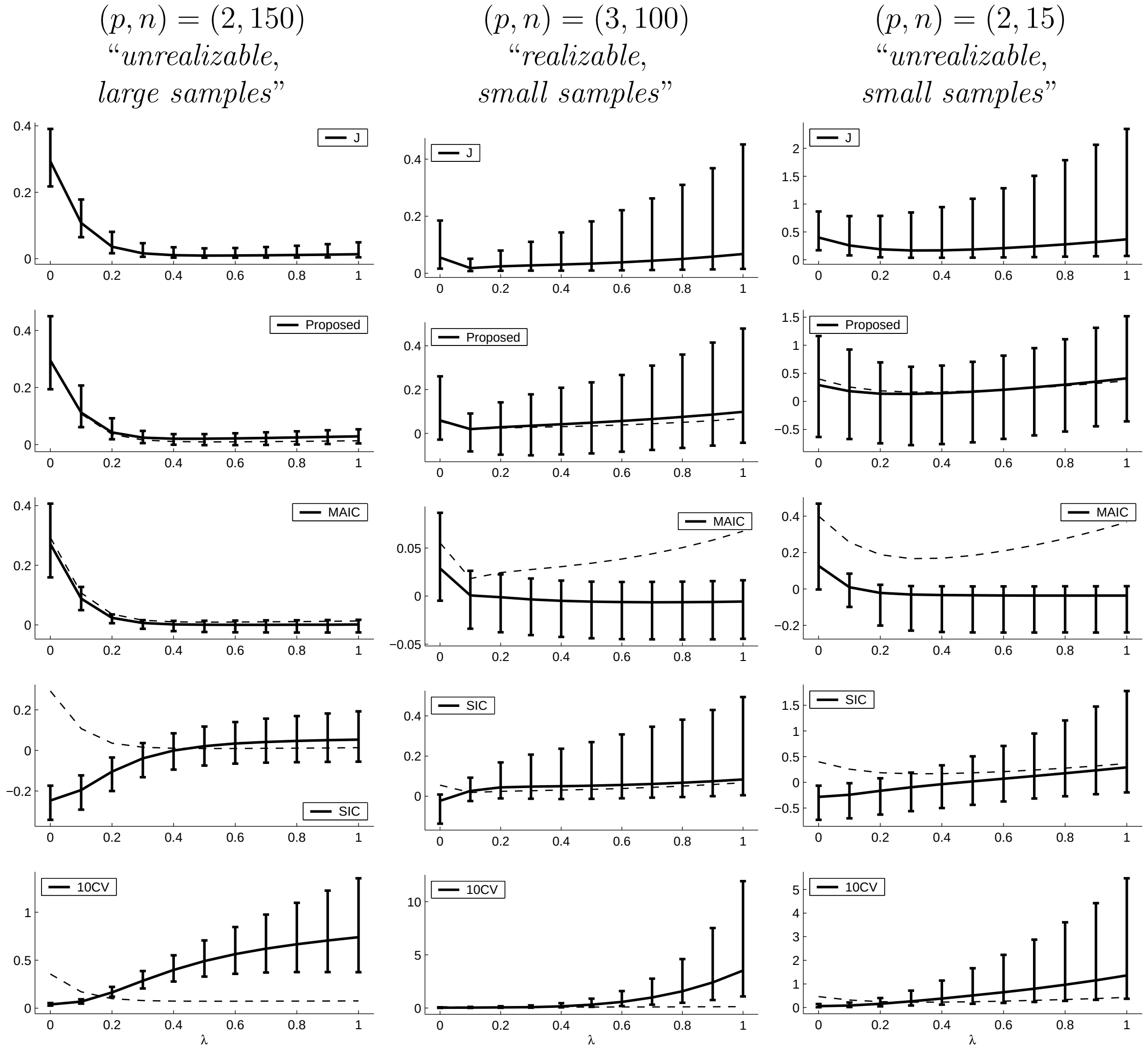
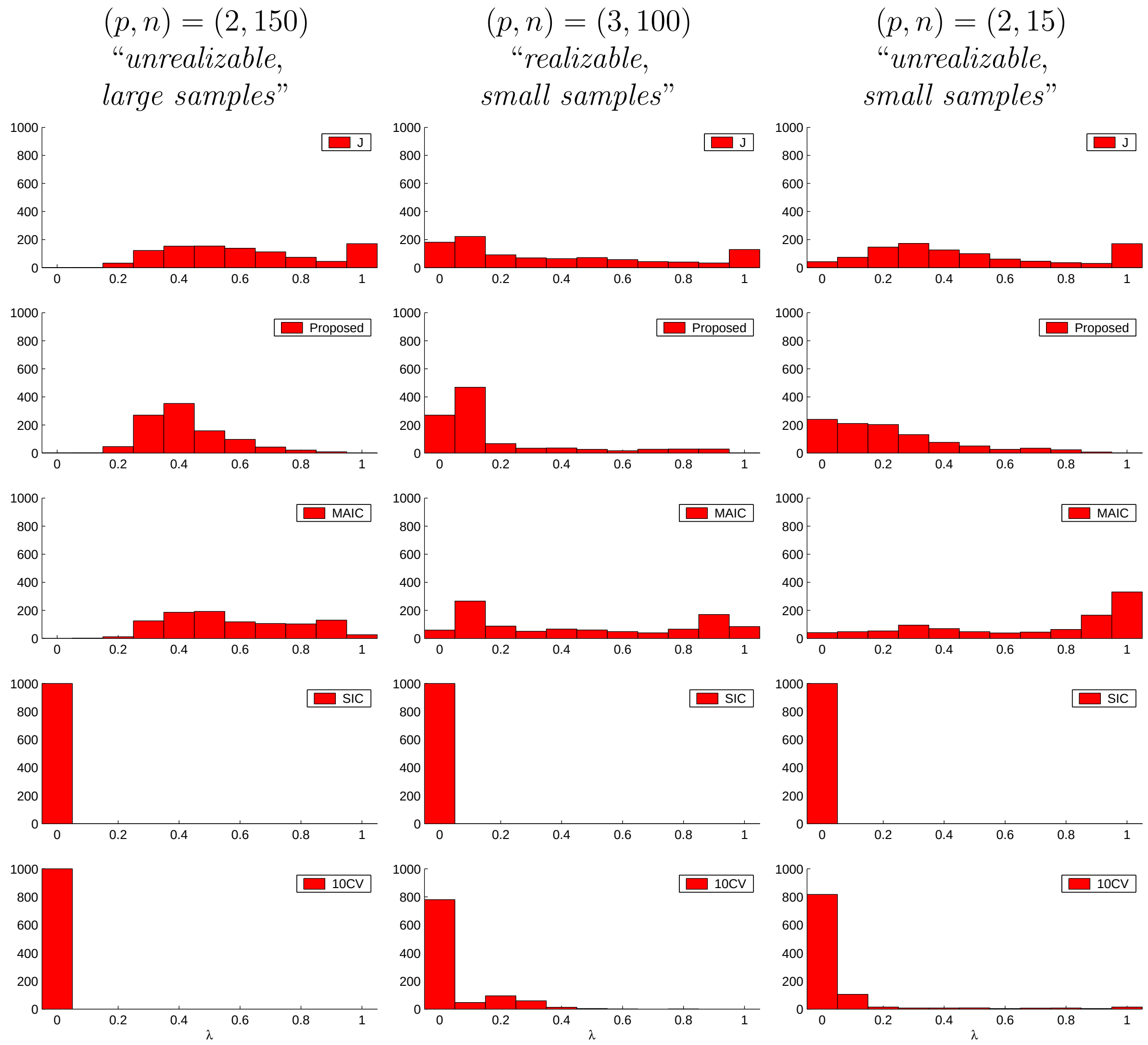


Figure 4: Extrapolation for the toy dataset. The mean and (asymmetric) standard deviation of each method are described as a function of the tuning parameter λ. The dashed curves in the bottom 12 graphs denote the mean of J. Each column corresponds to each (p, n).

for $\lambda = 0, 0.1, 0.2, \ldots, 1$. This procedure is repeated 1000 times for each (p, n). In the theoretical analysis, we fixed the training input points $\{x_i\}_{i=1}^{n}$ and only changed the noise $\{\epsilon_i\}_{i=1}^{n}$. On the other hand, we change both of them here because we are interested in investigating the accuracy of the methods for various different data.

Figure 4 depicts the mean and standard deviation of each method as a function of λ. Each column corresponds to each (p, n). Since the distribution was rather skewed, we calculated the lower and upper standard deviations separately. In order to make the comparison with J clear, we added the constant C (see Eq.(19)) to $\widehat{J}$, $\widehat{J}_{MAIC}$ and $\widehat{J}_{SIC}$



$(p,n) = (2,150)$ *"unrealizable, large samples"*

$(p,n) = (3,100)$ *"realizable, small samples"*

$(p,n) = (2,15)$ *"unrealizable, small samples"*

Figure 5: Extrapolation for the toy dataset. The distribution of λ chosen by each method is described.

because they are estimators of $J - C$. Similarly, we subtracted the constant σ^2 from 10CV because 10CV is an estimator of $J + \sigma^2$. The dashed curves in the bottom 12 graphs denote the mean of J.

When $(p,n) = (2,150)$, a large number of training examples are available so $\widehat{J}$ and $\widehat{J}_{MAIC}$ gave reasonably good unbiased estimates of the mean of J. $\widehat{J}_{SIC}$ was heavily biased because the realizability assumption is heavily violated. 10CV did not work properly because the assumption $p_x(x) = p_t(x)$ is not fulfilled. When $(p,n) = (3,100)$, $\widehat{J}$ and $\widehat{J}_{SIC}$ had reasonably good unbiasedness because the realizability assumption is roughly fulfilled. Note, however, that $\widehat{J}_{SIC}$ is rather inaccurate for very small λ, which we conjecture is caused by the slight violation of realizability. $\widehat{J}_{MAIC}$ was heavily biased since the number of training examples is rather small. 10CV did not work properly again because

Table 2: Extrapolation for the toy dataset. The mean and standard deviation of the test error obtained by each method are described. The best method and comparable ones by the t-test at the significance level 5% are described with boldface. For reference, the test error obtained with the optimal λ (i.e., the minimum test error) is described as 'OPT'.

(p,n)	$(2,150)$	$(3,100)$	$(2,15)$
OPT	0.06 ± 0.11	0.09 ± 0.14	0.91 ± 2.24
$\widehat{J}$	$\mathbf{0.15 \pm 0.23}$	$\mathbf{0.38 \pm 1.03}$	$\mathbf{2.69 \pm 5.18}$
MAIC	$\mathbf{0.13 \pm 0.19}$	0.51 ± 1.35	$\mathbf{3.23 \pm 8.13}$
SIC	2.93 ± 0.86	0.55 ± 0.79	4.00 ± 3.41
10CV	2.93 ± 0.86	0.47 ± 0.74	3.85 ± 3.45

All values are multiplied by 10 for compact description.

of $p_x(x) \neq p_t(x)$. Finally, in the challenging case of $(p,n) = (2,15)$ (i.e., "*unrealizable and small samples*"), the unbiasedness of $\widehat{J}$ was still reasonable, while the other methods were biased.

From these results, we can draw the following conclusions. When $(p,n) = (2,150)$ and $(3,100)$, the proposed $\widehat{J}$ is clearly shown to integrate the good properties of MAIC and SIC. This means that the primal goal of this paper has been surely accomplished. Furthermore, for the above toy example, $\widehat{J}$ seems to work better than other methods even in the challenging case of $(p,n) = (2,15)$.

Another interesting finding from the above results is that irrespective of (p,n), $\widehat{J}_{MAIC}$ has a tendency to reach the minimum at a large λ, while the minimizers of $\widehat{J}_{SIC}$ and 10CV tend to be small. On the other hand, the minimum of $\widehat{J}$ is adapted depending on (p,n) and seems to roughly agree with the minimum of J. These tendencies can also be observed in Figure 5, which shows the distribution of λ chosen by each method. Note, however, that the minimizer of $\widehat{J}$ tends to be slightly smaller than that of J.

Now we investigate the model selection performance. We chose the tuning parameter λ by each method, and estimated the output values for 100 test input points independently drawn from $p_t(x)$. The mean and standard deviation of the squared test error of each method over 1000 trials are described in Table 2. The best method and comparable ones by the *t-test* [15] at the significance level 5% are described with boldface. For reference, the test error obtained with the optimal λ (i.e., the minimum test error) is also described in the table as 'OPT'.

The table shows that when $(p,n) = (2,150)$, $\widehat{J}$ and $\widehat{J}_{MAIC}$ worked better than $\widehat{J}_{SIC}$ and 10CV. When $(p,n) = (3,100)$, $\widehat{J}$ outperformed other methods. We expected that $\widehat{J}_{SIC}$ also works well when $(p,n) = (3,100)$, but it did not. This implies that model selection by $\widehat{J}_{SIC}$ under covariate shift is not robust against the slight violation of the realizability assumption. Finally, when $(p,n) = (2,15)$, $\widehat{J}$ and $\widehat{J}_{MAIC}$ worked better than $\widehat{J}_{SIC}$ and 10CV. Although $\widehat{J}$ and $\widehat{J}_{MAIC}$ did not have significant difference by the t-test, the p-value was about 7%. Therefore, $\widehat{J}$ would be slightly better than $\widehat{J}_{MAIC}$. This result encourages us to use $\widehat{J}$ even in challenging scenarios of unrealizable and small sample cases.

We also performed similar simulations when $p_x(x)$ and $p_t(x)$ are unknown, but unlabeled samples $\{u_i\}_{i=1}^{100}$ which independently follow $p_t(x)$ are given in addition to the training examples $\{(x_i, y_i)\}_{i=1}^{n}$. We estimated $p_x(x)$ and $p_t(x)$ using $\{x_i\}_{i=1}^{n}$ and $\{u_i\}_{i=1}^{100}$ respectively by a kernel density estimation method with the Gaussian kernel and *Silverman's rule-of-thumb bandwidth selection rule* [36, 12]. That is, $\widehat{p}_x(x)$ was obtained as

$$\widehat{p}_x(x) = \frac{1}{n}\sum_{i=1}^{n}\phi_{x_i,h^2}(x), \tag{63}$$

where

$$h^2 = \left(\frac{4}{(d+2)n}\right)^{\frac{2}{d+4}}\widehat{\kappa}^2, \tag{64}$$

$$\widehat{\kappa}^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2, \tag{65}$$

$$\overline{x} = \sum_{j=1}^{n} x_j. \tag{66}$$

$\widehat{p}_t(x)$ was obtained similarly using the unlabeled samples $\{u_i\}_{i=1}^{100}$ instead of training input points $\{x_i\}_{i=1}^{n}$. Note, however, that we replaced $p_t(x)$ included in $\boldsymbol{U}$ (see Eq.(17)) not by $\widehat{p}_t(x)$ but by the empirical distribution of the unlabeled samples $\{u_i\}_{i=1}^{100}$ because it is computationally simple. It should also be noted that $p_x(x)$ and $p_t(x)$ included in the learning matrix $\boldsymbol{L}$ were also replaced by $\widehat{p}_x(x)$ and $\widehat{p}_t(x)$ (see Eq.(57)).

The simulation results obtained with $\widehat{p}_x(x)$ and $\widehat{p}_t(x)$ had similar tendency to the results obtained with $p_x(x)$ and $p_t(x)$ (for this reason, the graphs are omitted). From these results, we conjecture that the proposed method still works if $p_x(x)$ and $p_t(x)$ are estimated reasonably (cf. Section 3.5).

4.2 Multi-Dimensional Regression for Extrapolation

We also applied the proposed method to the *Abalone* data set available from the UCI repository [4]. It is a collection of 4177 samples, each of which consists of 8 input variables (physical measurements of abalones) and 1 output variable (the age of abalones). The first input variable is qualitative (male/female/infant) so it was ignored, and the other input variables were normalized to $[0, 1]$ for convenience. From the population, we randomly sampled n abalones for training and 100 abalones for testing. Here, we considered a biased sampling: the sampling has negative bias in the 4-th input variable (weight of abalones) for training and positive bias for testing. That is, the weight of training abalones tends to be small while that for the test abalones tends to be large[8]. Figure 6 depicts a realization

[8] More specifically, this was implemented as follows. A random number u is drawn from $N(0, (4177)^2)$ and let $v = \min(\lceil |u| \rceil, 4177)$. The v-th smallest abalone in the 4-th input variable is chosen for training.

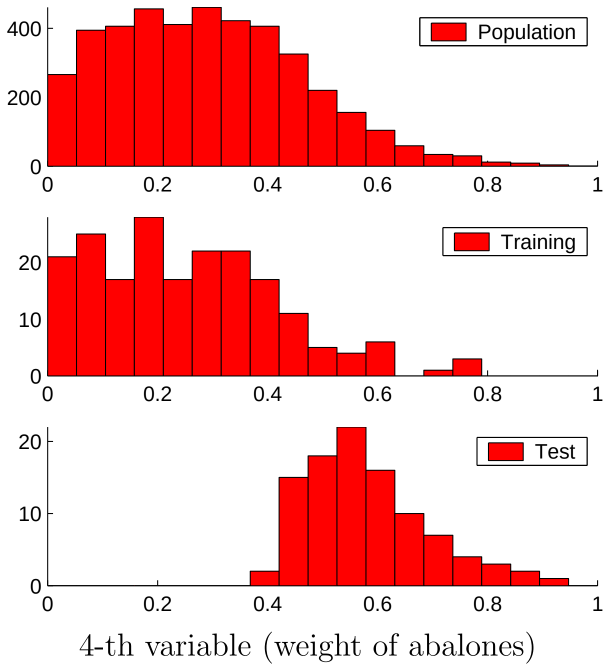


Figure 6: From top, the entire population of abalones, a realization of training abalones, and a realization of test abalones.

of the weight of training and test abalones when $n = 200$. We used multi-dimensional linear basis functions (i.e., the number of basis functions is $p = 8$) for learning. The density functions $p_x(\boldsymbol{x})$ and $p_t(\boldsymbol{x})$ were estimated using the same kernel density estimation method used in Section 4.1, where multi-dimensional Gaussian kernels without covariance are used and the variance of the Gaussian kernel is determined by using Eqs.(64)–(66) in a coordinate-wise manner. A notable difference from the experiments in Section 4.1 is that we used the test input points themselves for estimating $p_t(\boldsymbol{x})$, not unlabeled samples. Therefore, the setting corresponds to the *transductive inference* [49].

Figure 7 depicts the mean values of each method over 300 trials for $n = 50, 200$, and 800. The error bars are omitted because they were excessive and deteriorated the graphs. Note that the true generalization error J was calculated using the test examples. The 3 graphs in the top row show that the best λ surely increases as n gets large, as stated in the beginning of this section. The proposed $\widehat{J}$ seems to give reasonably good curves and its minimum roughly agrees with the minimum of the true test error (see the second row). On the other hand, irrespective of n, the minimizer of $\widehat{J}_{MAIC}$ tends to be large and the minimizers of $\widehat{J}_{SIC}$ and 10CV tend to be small (see the third to fifth rows). This result is consistent with the previous one-dimensional simulations. Similar tendencies can be observed in Figure 8, which depicts the distribution of λ chosen by each method.

Another important finding from the graphs in Figure 7 is that the magnitude of the

This is repeated until n abalones are selected without overlapping. Then, from the rest, 100 test abalones are chosen similarly, using u drawn from $N(0, (4177/10)^2)$ and $v = 4177 - \min(\lceil |u| \rceil, 4177) + 1$.

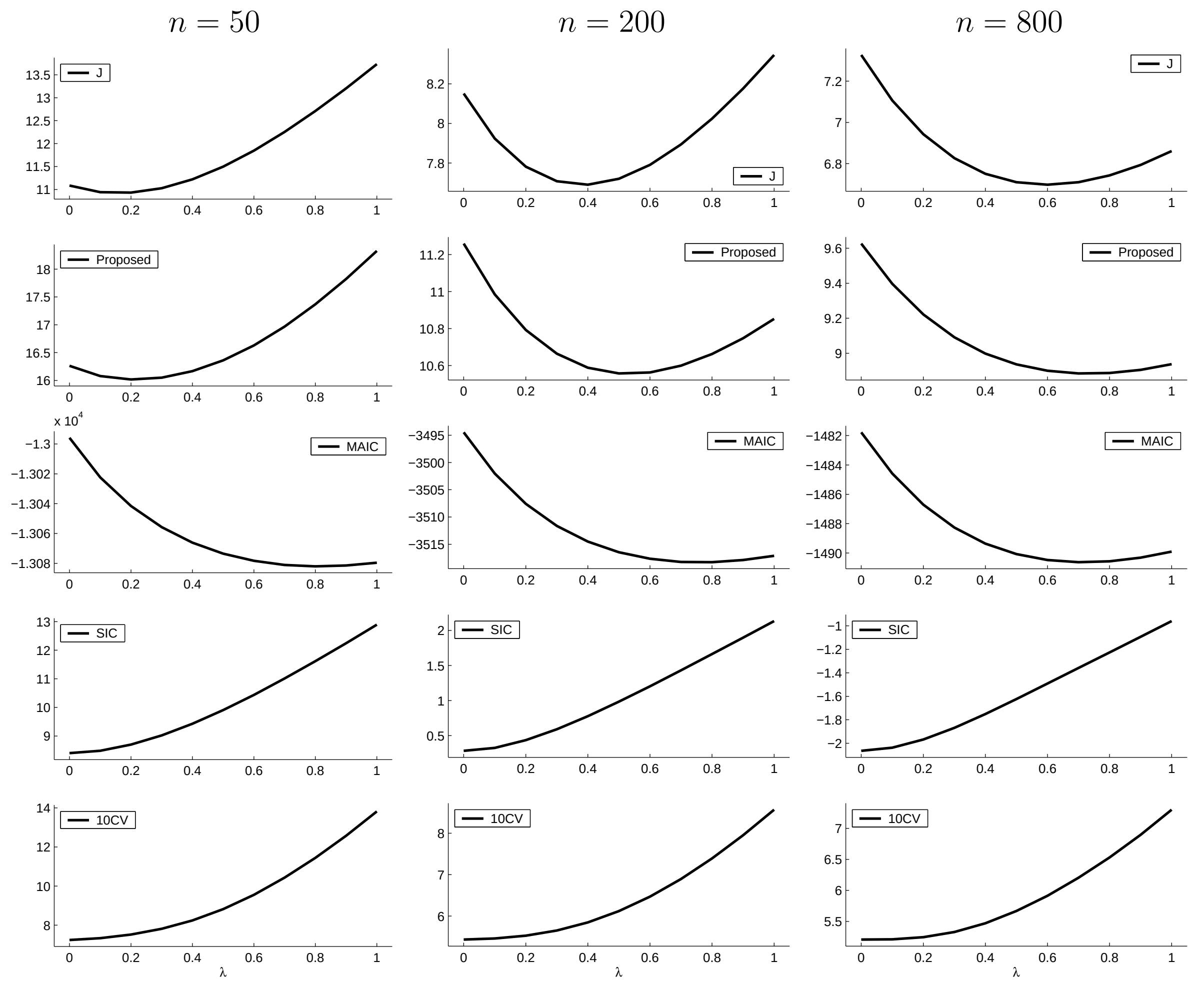


Figure 7: Extrapolation of the 4-th variable for the Abalone dataset. The mean of each method is described. Each column corresponds to each n.

values of $\widehat{J}_{MAIC}$ is very large (see the third row), which may be explained as follows. The value $p_t(\boldsymbol{x}_i)/p_x(\boldsymbol{x}_i)$ can be very large in multidimensional cases because the input domain $\mathcal{D}$ is so vast that the values of $p_x(\boldsymbol{x})$ tend to be very small. Then the magnitude of the elements of matrix $\widehat{\boldsymbol{U}}$ included in $\widehat{J}_{MAIC}$ can also become huge, which makes the magnitude of $\widehat{J}_{MAIC}$ very large. Note that $p_t(\boldsymbol{x}_i)/p_x(\boldsymbol{x}_i)$ also appears in $\widehat{J}$ via $\widehat{\boldsymbol{L}}_u$. However, $\widehat{\boldsymbol{L}}_u$ does not cause such problems because it also includes the inverse of the above quantity so it is balanced. Thus, the proposed $\widehat{J}$ appears also more reliable than $\widehat{J}_{MAIC}$.

We chose the tuning parameter λ by each method and estimated the age of the test abalones by using the chosen λ. The mean squared test error for all test abalones were calculated, and this procedure was repeated 300 times. The mean and standard deviation of the test error of each method over 300 trials are described in Table 3, showing that $\widehat{J}$ gave small errors for all cases. On the other hand, the error obtained by $\widehat{J}_{MAIC}$ was

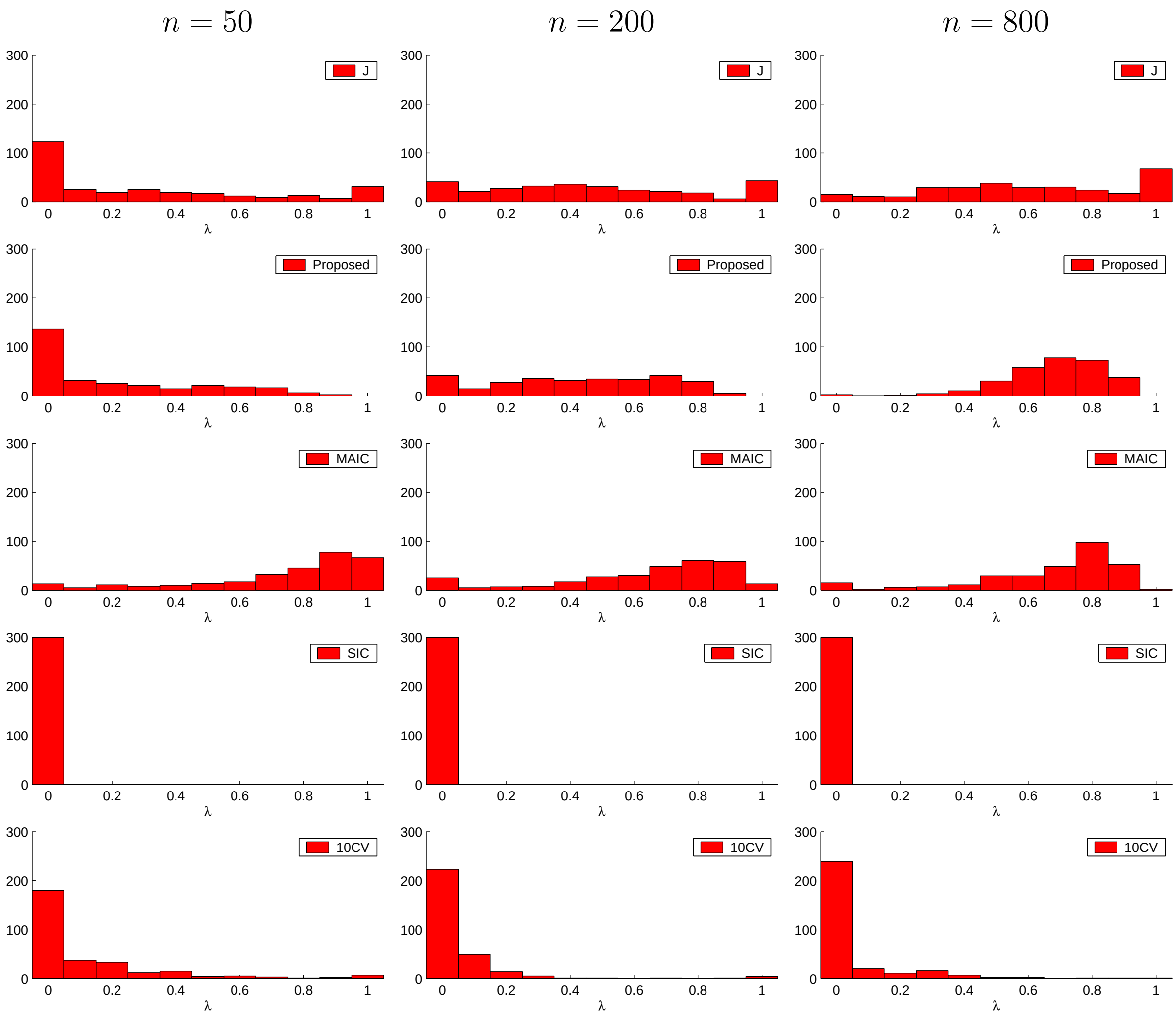


Figure 8: Extrapolation of the 4-th variable for the Abalone dataset. The distribution of λ chosen by each method is described.

large when n is small, and $\widehat{J}_{SIC}$ and 10CV gave large errors when n is large. Hence, the proposed method overall compares favorably with the other methods. However, the remaining gap between the proposed method and the optimal choice especially in small sample cases implies that there is still room for improvement.

We also carried out similar simulations when the sampling is biased in the 6-th input variable (weight of gut after bleeding). The results described in Table 4 show similar trends to the previous ones.

4.3 Binary Classification with Imbalanced Data

Finally, let us consider binary classification problems from imbalanced training examples. More specifically, we consider the cases where the number of training examples for the

Table 3: Extrapolation of the 4-th variable in the Abalone dataset. The mean and standard deviation of the test error obtained with each method are described. The best method and comparable ones by the t-test at the significance level 5% are described with boldface. For reference, the test error obtained with the optimal λ (i.e., the minimum test error) is described as 'OPT'.

n	50	200	800
OPT	9.86 ± 4.27	7.40 ± 1.77	6.54 ± 1.34
$\widehat{J}$	$\mathbf{11.67 \pm 5.74}$	$\mathbf{7.95 \pm 2.15}$	$\mathbf{6.77 \pm 1.40}$
MAIC	12.78 ± 6.71	$\mathbf{8.01 \pm 2.27}$	$\mathbf{6.77 \pm 1.42}$
SIC	$\mathbf{11.09 \pm 5.23}$	$\mathbf{8.15 \pm 1.95}$	7.33 ± 1.37
10CV	$\mathbf{10.88 \pm 5.05}$	$\mathbf{8.06 \pm 1.91}$	7.23 ± 1.37

Table 4: Extrapolation of the 6-th variable in the Abalone dataset.

n	50	200	800
OPT	9.04 ± 4.04	6.76 ± 1.68	6.05 ± 1.25
$\widehat{J}$	$\mathbf{10.67 \pm 6.19}$	$\mathbf{7.31 \pm 2.24}$	$\mathbf{6.20 \pm 1.33}$
MAIC	11.16 ± 7.02	$\mathbf{7.23 \pm 2.07}$	$\mathbf{6.20 \pm 1.32}$
SIC	$\mathbf{10.30 \pm 4.74}$	$\mathbf{7.46 \pm 1.81}$	6.76 ± 1.27
10CV	$\mathbf{10.15 \pm 4.95}$	$\mathbf{7.42 \pm 1.81}$	6.68 ± 1.25

positive class is significantly larger than that for the negative class while the ratio of samples in both classes is even for test examples.

For such imbalanced data, it seems common particularly in the neural network community to increase the "influence" of the training examples for the minor class so that the influence of the training examples are balanced [21, 5]. The weighted learning scheme given by Eq.(56) actually implements such balancing automatically. Therefore, we will use the same weighted least-squares learning here. Note that in classification scenarios, it may be more natural to use loss functions such as the *hinge loss* [32] rather than to use the squared loss. However, it is claimed from experiments that classification with the squared loss works as well as that with the hinge loss [8, 45]. For this reason, we decided to use the squared loss for learning here.

Let us denote the probability density functions of the positive and negative classes by $p_+(\boldsymbol{x})$ and $p_-(\boldsymbol{x})$, respectively. In this experiment, we put the input dimension $d = 2$ and

$$p_+(\boldsymbol{x}) = \phi_{(2,0)^\top, 2\boldsymbol{I}}(\boldsymbol{x}), \tag{67}$$

$$p_-(\boldsymbol{x}) = \phi_{(-2,0)^\top, 2\boldsymbol{I}}(\boldsymbol{x}). \tag{68}$$

Let the training and test input densities be

$$p_x(\boldsymbol{x}) = 0.9p_+(\boldsymbol{x}) + 0.1p_-(\boldsymbol{x}), \tag{69}$$

$$p_t(\boldsymbol{x}) = 0.5p_+(\boldsymbol{x}) + 0.5p_-(\boldsymbol{x}). \tag{70}$$

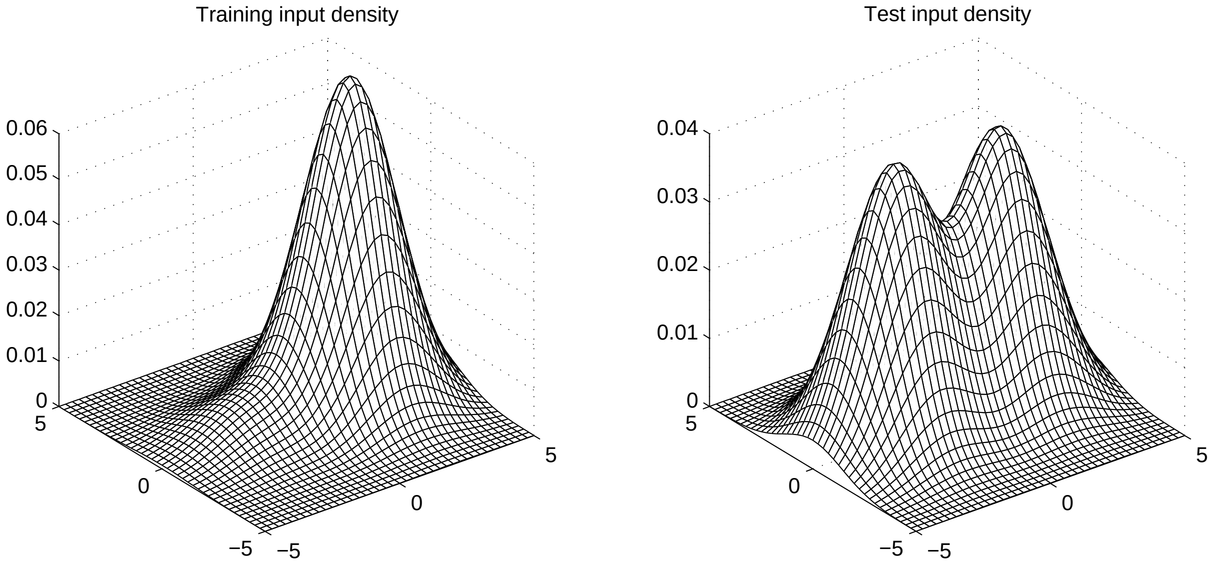


Figure 9: Training input density function $p_x(\boldsymbol{x})$ (left) and test input density function $p_t(\boldsymbol{x})$ (right) for binary classification with imbalanced data.

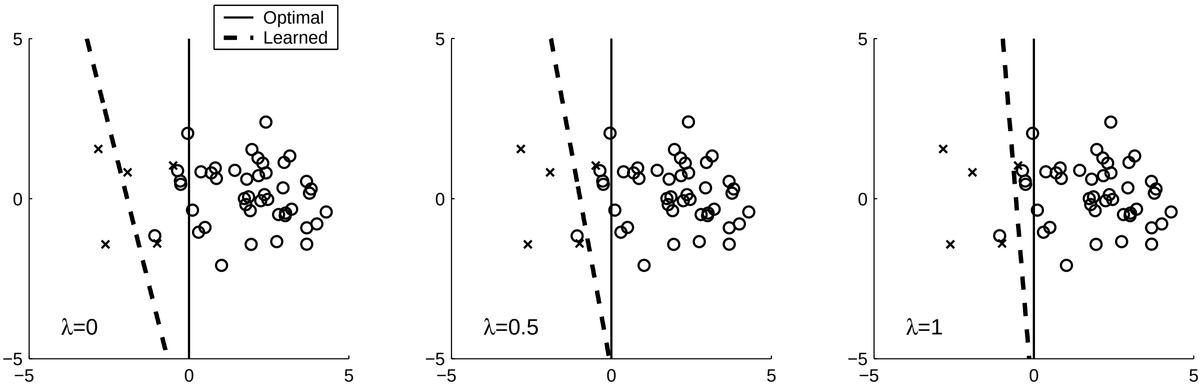


Figure 10: Examples of learned decision boundaries for different λ. $\times$'s and $\circ$'s denote the training examples for the negative and positive classes, respectively. The solid line denotes the optimal decision boundary while the dashed line denotes the learned decision boundaries.

These density functions are depicted in Figure 9. We created n training examples and 5000 test examples following $p_x(\boldsymbol{x})$ and $p_t(\boldsymbol{x})$, respectively. Multi-dimensional linear basis functions (i.e., the number of basis functions is $p = 3$) are used for learning.

Examples of learned decision boundaries for $n = 100$ are depicted in Figure 10. When $\lambda = 0$, the learned decision boundary was close to the negative examples ($\times$'s) because the influence of the negative examples is too weak. As λ increased, the decision boundary was shifted toward positive examples ($\circ$'s). This happened because $p_x(\boldsymbol{x}_i)$ is small for negative examples (see Eq.(56)) so the influence of the negative examples tends to be

Table 5: Misclassification rate for test samples in two-dimensional classification problem with imbalanced data.

n	50	100	200
OPT	15.10 ± 8.28	13.30 ± 4.78	12.31 ± 2.94
$\widehat{J}$	$\mathbf{16.00 \pm 8.62}$	$\mathbf{13.85 \pm 4.96}$	$\mathbf{12.63 \pm 3.04}$
MAIC	16.87 ± 10.19	14.34 ± 5.93	$\mathbf{12.86 \pm 3.42}$
10CV	19.63 ± 10.89	17.16 ± 7.38	15.64 ± 4.50

emphasized. In this example, $\lambda = 1$ gave the best result among three cases.

We calculated the values of $\widehat{J}$, $\widehat{J}_{MAIC}$, and 10CV as a function of $\lambda = 0, 0.1, 0.2, \ldots, 1$. Note that in this experiment, the true generalization error was measured by the misclassification rate for the test samples, i.e., we used the *0/1-loss function* for the generalization error. To be consistent with this generalization error, CV was also calculated using the 0/1-loss. On the other hand, $\widehat{J}$ was calculated using the squared loss because it can not deal with the 0/1-loss (see Eq.(13)). Therefore, applying the proposed method to classification tasks is a heuristic. Note that $\widehat{J}_{MAIC}$ was also calculated with the squared loss (or equivalently, Gaussian noise model). We chose the tuning parameter λ by each method, and calculated the misclassification rate for the test samples by using the chosen λ. For each $n = 50$, 100, and 200, this procedure was repeated 1000 times. When one of the classes had no training examples, we redrew the training examples until each class had at least one training example. Note that in this simulation, the minimum of the CV score was often not unique because the CV score is discrete due to the 0/1-loss function and thus takes the same value for different λ's. In such cases, we randomly chose one of the best λ's.

The mean and standard deviation of the obtained misclassification rate are described in Table 5, showing that the proposed method significantly outperforms cross-validation and is better than the modified AIC especially in small sample cases. This simulation result implies that the proposed method is practically useful even for imbalanced classification tasks.

5 Conclusions and Discussion

In this paper, we proposed a new generalization error estimation method when the training and test distributions are different. It can effectively integrate the advantages of the modified AIC and SIC, i.e., it is (almost) unbiased with finite samples in (almost) realizable cases and asymptotically unbiased in general (see Table 1). The numerical evaluations in extrapolation scenarios (Figure 4) showed that (a) the proposed method works well both in the case of realizable and small samples and in the case of unrealizable and large samples, and (b) it provided promising performance even in a challenging case of unrealizable and small samples. While it was experimentally observed that the modified AIC can be unstable for high-dimensional data, our method is more stable (Figure 7).

Furthermore our method also worked excellently in a classification task with imbalanced data.

The proposed generalization error estimator $\widehat{J}$ can be actually regarded as an extension of SIC by the following fact. We can prove that the exact unbiasedness of SIC in realizable cases holds not only for $\widetilde{\boldsymbol{L}}_u$ given by Eq.(48), but also for any matrix of the form

$$(\boldsymbol{X}^\top \boldsymbol{X})^{-1}\boldsymbol{X}^\top + \boldsymbol{Z}(\boldsymbol{I} - \boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{X})^{-1}\boldsymbol{X}^\top), \tag{71}$$

where $\boldsymbol{Z}$ is an arbitrary $p \times n$ matrix. If we put $\boldsymbol{Z} = (\boldsymbol{X}^\top \boldsymbol{D}\boldsymbol{X})^{-1}\boldsymbol{X}^\top \boldsymbol{D}$, the above matrix is reduced to $\widehat{\boldsymbol{L}}_u$ and $\widehat{J}$ is obtained. From this fact, the contributions of this paper can be interpreted as follows. We pointed out that SIC is applicable to cases where the distributions of training and test input points are different. We showed that for achieving the exact unbiasedness in realizable cases, there exists a degree of freedom in the choice of $\boldsymbol{Z}$ in SIC and that by appropriately choosing $\boldsymbol{Z}$, the asymptotic unbiasedness in unrealizable cases can be gained in addition to the exact unbiasedness in realizable cases. We further found that $\widehat{J}$ is useful for estimating the difference of the generalization error, which is an important theoretical property in the context of model selection (see Section 3.3).

Although we focused on the cases where the training and test distributions are different, the analyses given in this paper are valid even when they are equivalent. Therefore, as long as we can reasonably estimate the input density function using, e.g., a large number of unlabeled samples, the proposed $\widehat{J}$ is still effective in model comparison even when the training and test distributions are common.

So far we restricted ourselves to the cases where the parameters in the regression model are learned in a linear manner (see Eq.(6)). However, there are useful learning methods which are non-linear, e.g., learning with non-quadratic loss functions [16, 53, 49] or non-quadratic regularizers [10, 52, 48, 6]. Extending the current approach to be able to deal with such non-linear learning methods is an important future direction. It would furthermore be interesting to investigate whether similar generalization error estimators can be derived for non-squared test errors, e.g., for the misclassification rate.

In Section 3.3, we investigated the asymptotic effectiveness of the proposed generalization error estimator in model comparison. There, the asymptotic effectiveness in model comparison was evaluated in terms of the asymptotic upper bounds. Carrying out more precise analysis, e.g., in terms of the exact asymptotic order is a remaining future work.

The numerical simulations showed that the proposed method is a reasonably accurate unbiased estimator. However, it can have a large variance especially when the number of training examples is very small or the noise level is very high. This implies that our ultimate goal is not to estimate the generalization error in an unbiased manner, but to accurately estimate it for a single realization. For realizable cases, a method to "regularize" unbiased generalization error estimators has been recently proposed [39], which yielded a more accurate estimator for a single realization. It is interesting to investigate whether a similar or novel strategy can work also for unrealizable cases. Furthermore, in the context of model selection, it is important to investigate the effectiveness in model comparison (see Section 3.3) not only in terms of the expectation but also in terms of a single realization. Finally, it is important to theoretically investigate the model selection performance using the proposed generalization error estimator, e.g., following the idea of the reference [22].

Acknowledgments

The authors would like to thank Dr. Motoaki Kawanabe and Dr. Gilles Blanchard of Fraunhofer FIRST for the fruitful discussions. Special thanks also go to anonymous reviewers for their valuable comments. We acknowledge partial financial support from the Alexander von Humboldt Foundation, from DFG (# Mu 987/2-1 and # JA 379/13-2), from the PASCAL Network of Excellence (EU #506778) and from MEXT (Grant-in-Aid for Young Scientists 17700142).

A Proof of Lemma 1

From Eqs.(28) and (18), Eq.(29) yields

$$B_{\boldsymbol{\epsilon}} = -2\mathbb{E}_{\boldsymbol{\epsilon}}\langle \boldsymbol{U}\boldsymbol{L}\boldsymbol{y}, \widehat{\boldsymbol{L}}_u\boldsymbol{y} - \boldsymbol{\alpha}^*\rangle + 2\mathbb{E}_{\boldsymbol{\epsilon}}\widehat{\sigma_u^2}\mathrm{tr}(\boldsymbol{U}\boldsymbol{L}\widehat{\boldsymbol{L}}_u^\top). \tag{72}$$

Let $\boldsymbol{z}_f$, $\boldsymbol{z}_g$ and $\boldsymbol{z}_r$ be n-dimensional vectors with i-th elements $f(\boldsymbol{x}_i)$, $g(\boldsymbol{x}_i)$, and $r(\boldsymbol{x}_i)$, respectively. Then we have $\boldsymbol{z}_g = \boldsymbol{X}\boldsymbol{\alpha}^*$, $\boldsymbol{\alpha}^* = \widehat{\boldsymbol{L}}_u\boldsymbol{z}_g$, and $\mathbb{E}_{\boldsymbol{\epsilon}}\widehat{\sigma_u^2} = \sigma^2 + \zeta$, where $\zeta = \|\boldsymbol{G}\boldsymbol{z}_r\|^2/\mathrm{tr}(\boldsymbol{G})$. From them, we have

$$B_{\boldsymbol{\epsilon}} = -2\langle \boldsymbol{U}\boldsymbol{L}\boldsymbol{z}_f, \widehat{\boldsymbol{L}}_u\boldsymbol{z}_r\rangle + 2\zeta\mathrm{tr}(\boldsymbol{U}\boldsymbol{L}\widehat{\boldsymbol{L}}_u^\top), \tag{73}$$

from which Eqs.(30) and (31) are clear. In the following, we investigate the asymptotic order of each term.

By noting that training input points $\{\boldsymbol{x}_i\}_{i=1}^n$ independently follows the probability distribution with the probability density function $p_x(\boldsymbol{x})$ and by using the law of large numbers [29], we have

$$\begin{aligned} \lim_{n\to\infty}\left(\frac{1}{n}\sum_{k=1}^n \frac{p_t(\boldsymbol{x}_k)}{p_x(\boldsymbol{x}_k)}\varphi_i(\boldsymbol{x}_k)\varphi_j(\boldsymbol{x}_k)\right) &= \int_{\mathcal{D}} \frac{p_t(\boldsymbol{x})}{p_x(\boldsymbol{x})}\varphi_i(\boldsymbol{x})\varphi_j(\boldsymbol{x})p_x(\boldsymbol{x})d\boldsymbol{x} \\ &= \langle \varphi_i, \varphi_j\rangle_{\mathcal{H}}, \end{aligned} \tag{74}$$

implying that $\frac{1}{n}\boldsymbol{X}^\top\boldsymbol{D}\boldsymbol{X} = \mathcal{O}_p(1)$. Since $\frac{1}{n}\boldsymbol{X}^\top\boldsymbol{D}\boldsymbol{X}$ is invertible by the assumption (5), we have $(\frac{1}{n}\boldsymbol{X}^\top\boldsymbol{D}\boldsymbol{X})^{-1} = \mathcal{O}_p(1)$. Furthermore, by the central limit theorem [29], it holds for sufficiently large n,

$$\begin{aligned} \frac{1}{n}\sum_{i=1}^n r(\boldsymbol{x}_i)\varphi_j(\boldsymbol{x}_i)\frac{p_t(\boldsymbol{x}_i)}{p_x(\boldsymbol{x}_i)} &= \int_{\mathcal{D}} r(\boldsymbol{x})\varphi_j(\boldsymbol{x})\frac{p_t(\boldsymbol{x})}{p_x(\boldsymbol{x})}p_x(\boldsymbol{x})d\boldsymbol{x} + \mathcal{O}_p(n^{-\frac{1}{2}}) \\ &= \langle r, \varphi_j\rangle_{\mathcal{H}} + \mathcal{O}_p(n^{-\frac{1}{2}}) = \mathcal{O}_p(n^{-\frac{1}{2}}), \end{aligned} \tag{75}$$

implying that $\frac{1}{n}\boldsymbol{X}^\top\boldsymbol{D}\boldsymbol{z}_r = \mathcal{O}_p(n^{-\frac{1}{2}})$. Therefore, we have

$$\widehat{\boldsymbol{L}}_u\boldsymbol{z}_r = (\tfrac{1}{n}\boldsymbol{X}^\top\boldsymbol{D}\boldsymbol{X})^{-1}\tfrac{1}{n}\boldsymbol{X}^\top\boldsymbol{D}\boldsymbol{z}_r = \mathcal{O}_p(n^{-\frac{1}{2}}). \tag{76}$$

On the other hand, it holds that $\boldsymbol{U}\boldsymbol{L}\boldsymbol{z_f} = \mathcal{O}_p(1)$, $\zeta = \mathcal{O}_p(1)$, and $\mathrm{tr}(\boldsymbol{U}\boldsymbol{L}\widehat{\boldsymbol{L}}_u^\top) = \mathcal{O}_p(n^{-1})$, from which we have Eq.(32). (Q.E.D.)

Remark: Let us consider non-parametric cases where p increases as n increases. We assume that $(\frac{1}{n}\boldsymbol{X}^\top\boldsymbol{D}\boldsymbol{X})^{-1} = O_p(p^{-1})$, which may not be so restrictive since $\frac{1}{n}\boldsymbol{X}^\top\boldsymbol{D}\boldsymbol{X} = O_p(1)$. Then, even in non-parametric cases, we still have $\widehat{\boldsymbol{L}}_u\boldsymbol{z}_r = \mathcal{O}_p(n^{-\frac{1}{2}})$. By the central limit theorem, we have $\boldsymbol{U} = \frac{1}{n}\boldsymbol{X}^\top\boldsymbol{D}\boldsymbol{X} + \Delta\boldsymbol{U}$ where $\Delta\boldsymbol{U} = \mathcal{O}_p(n^{-\frac{1}{2}})$. Then we have

$$\boldsymbol{U}\widehat{\boldsymbol{L}}_u\boldsymbol{z}_r = \tfrac{1}{n}\boldsymbol{X}^\top\boldsymbol{D}\boldsymbol{z}_r + \Delta\boldsymbol{U}\widehat{\boldsymbol{L}}_u\boldsymbol{z}_r = \mathcal{O}_p(n^{-\frac{1}{2}}) + \mathcal{O}_p(pn^{-1}). \tag{77}$$

Since $\boldsymbol{L}\boldsymbol{z}_f = \mathcal{O}_p(1)$, we have

$$\langle \boldsymbol{L}\boldsymbol{z}_f, \boldsymbol{U}\widehat{\boldsymbol{L}}_u\boldsymbol{z}_r\rangle = \mathcal{O}_p(pn^{-\frac{1}{2}}) + \mathcal{O}_p(p^2n^{-1}). \tag{78}$$

Since $\boldsymbol{G}$ is a projection matrix onto a $(n-p)$-dimensional subspace, we have $\mathrm{tr}(\boldsymbol{G}) = n-p$ and $\|\boldsymbol{G}\boldsymbol{z}_r\|^2 \le \|\boldsymbol{z}_r\|^2 = \mathcal{O}_p(n)$, which implies $\zeta = \mathcal{O}_p(1)$ because of Eq.(3). Since $\boldsymbol{L}\widehat{\boldsymbol{L}}_u^\top$ is a p-dimensional matrix of $\mathcal{O}_p(n^{-1})$, we have $\mathrm{tr}(\boldsymbol{U}\boldsymbol{L}\widehat{\boldsymbol{L}}_u^\top) = \mathcal{O}_p(p^2n^{-1})$. Therefore, we have

$$\zeta\mathrm{tr}(\boldsymbol{U}\boldsymbol{L}\widehat{\boldsymbol{L}}_u^\top) = \mathcal{O}_p(p^2n^{-1}). \tag{79}$$

From Eqs.(78) and (79), we have

$$B_{\boldsymbol{\epsilon}} = \mathcal{O}_p(pn^{-\frac{1}{2}}) + \mathcal{O}_p(p^2n^{-1}), \tag{80}$$

which is $o_p(1)$ if $p = o(n^{\frac{1}{2}})$.

B Proof of Lemma 4

From Eq.(73), we have

$$\Delta B_{\boldsymbol{\epsilon}} = -2\langle \boldsymbol{U}\Delta\boldsymbol{L}\boldsymbol{z}_f, \widehat{\boldsymbol{L}}_u\boldsymbol{z}_r\rangle + 2\zeta\mathrm{tr}(\boldsymbol{U}\Delta\boldsymbol{L}\widehat{\boldsymbol{L}}_u^\top). \tag{81}$$

From Eq.(40), we have $\Delta\boldsymbol{L}\boldsymbol{z_f} = \mathcal{O}_p(n^{-t+1})$. Then $\boldsymbol{U} = \mathcal{O}(1)$ and Eq.(76) yield $\langle \boldsymbol{U}\Delta\boldsymbol{L}\boldsymbol{z}_f, \widehat{\boldsymbol{L}}_u\boldsymbol{z}_r\rangle = \mathcal{O}_p(n^{-t+\frac{1}{2}})$, and $\zeta = \mathcal{O}_p(1)$ and $\widehat{\boldsymbol{L}}_u = \mathcal{O}_p(n^{-1})$ yield $\zeta\mathrm{tr}(\boldsymbol{U}\Delta\boldsymbol{L}\widehat{\boldsymbol{L}}_u^\top) = \mathcal{O}_p(n^{-t})$. Therefore we have $\Delta B_{\boldsymbol{\epsilon}} = \mathcal{O}_p(n^{-(t-\frac{1}{2})})$.

On the other hand, from Eq.(18), we have

$$\mathbb{E}_{\boldsymbol{\epsilon}}[J] = \langle \boldsymbol{U}\boldsymbol{L}\boldsymbol{z}_f, \boldsymbol{L}\boldsymbol{z}_f\rangle + \sigma^2\mathrm{tr}(\boldsymbol{U}\boldsymbol{L}\boldsymbol{L}^\top) - 2\langle \boldsymbol{U}\boldsymbol{L}\boldsymbol{z}_f, \boldsymbol{\alpha}^*\rangle + C, \tag{82}$$

from which we have

$$\mathbb{E}_{\boldsymbol{\epsilon}}[\Delta J] = \langle \boldsymbol{U}\Delta\boldsymbol{L}\boldsymbol{z}_f, \boldsymbol{b}\rangle + \sigma^2 \mathrm{tr}(\boldsymbol{U}(\boldsymbol{L}_1 + \boldsymbol{L}_2)\Delta\boldsymbol{L}^\top). \tag{83}$$

where $\boldsymbol{b} = (\boldsymbol{L}_1 + \boldsymbol{L}_2)\boldsymbol{z}_f - 2\boldsymbol{\alpha}^*$. Since $\boldsymbol{b} = \mathcal{O}_p(n^{-u})$, we have $\mathbb{E}_{\boldsymbol{\epsilon}}[\Delta J] = \mathcal{O}_p(n^{-t+1-u}) + \mathcal{O}_p(n^{-t}) = \mathcal{O}_p(n^{-(t-1+u)})$. (Q.E.D.)

Remark: Let us consider non-parametric cases where p increases as n increases. Suppose again $(\frac{1}{n}\boldsymbol{X}^\top\boldsymbol{D}\boldsymbol{X})^{-1} = O_p(p^{-1})$. From Eq.(77), we have

$$\begin{aligned} \Delta B_{\boldsymbol{\epsilon}} &= -2\langle \Delta\boldsymbol{L}\boldsymbol{z}_f, \boldsymbol{U}\widehat{\boldsymbol{L}}_u\boldsymbol{z}_r\rangle + 2\zeta\mathrm{tr}(\boldsymbol{U}\Delta\boldsymbol{L}\widehat{\boldsymbol{L}}_u^\top) \\ &= \mathcal{O}_p(n^{-t+\frac{1}{2}}p) + \mathcal{O}_p(n^{-t}p^2). \end{aligned} \tag{84}$$

On the other hand,

$$\mathbb{E}_{\boldsymbol{\epsilon}}[\Delta J] = \mathcal{O}_p(n^{-t+1-u}p^2) + \mathcal{O}_p(n^{-t}p^2) = \mathcal{O}_p(n^{-t+1-u}p^2), \tag{85}$$

which implies the asymptotic effectiveness of $\widehat{J}$ in model comparison.

C Proof of Lemma 5

$p_t(\boldsymbol{x})$ is included in $\boldsymbol{U}$ and $\widehat{\boldsymbol{L}}_u$. Let $\boldsymbol{U}'$ and $\widehat{\boldsymbol{L}}_u'$ be $\boldsymbol{U}$ and $\widehat{\boldsymbol{L}}_u$ calculated with $\widehat{p}_t(\boldsymbol{x})$ instead of $p_t(\boldsymbol{x})$. It is clear that $\boldsymbol{U}' = \boldsymbol{U} + \mathcal{O}(\xi_t)$. For a nonsingular symmetric matrix $\boldsymbol{T}$ and a matrix $\boldsymbol{B}$, it holds that $(\boldsymbol{T} + \eta\boldsymbol{B})^{-1} = \boldsymbol{T}^{-1} + \mathcal{O}(\eta)$ for sufficiently small η [2], from which we have $\widehat{\boldsymbol{L}}_u' = \widehat{\boldsymbol{L}}_u + \mathcal{O}(\eta_t)$. This implies Eq.(51). (Q.E.D.)

D Proof of Lemma 6

$p_x(\boldsymbol{x})$ is included only in $\widehat{\boldsymbol{L}}_u$. Let $\widehat{\boldsymbol{L}}_u''$ be $\widehat{\boldsymbol{L}}_u$ calculated with $\widehat{p}_x(\boldsymbol{x})$ instead of $p_x(\boldsymbol{x})$. It holds that

$$\left|\frac{1}{\widehat{p}_x(\boldsymbol{x}_i)} - \frac{1}{p_x(\boldsymbol{x}_i)}\right| = \left|\frac{\widehat{p}_x(\boldsymbol{x}_i) - p_x(\boldsymbol{x}_i)}{p_x(\boldsymbol{x}_i)\widehat{p}_x(\boldsymbol{x}_i)}\right| \leq \frac{\eta_x}{\gamma\widehat{\gamma}}. \tag{86}$$

Therefore, by a similar discussion to the proof of Lemma 5, we have $\widehat{\boldsymbol{L}}_u'' = \widehat{\boldsymbol{L}}_u + \mathcal{O}(\frac{\eta_x}{\gamma\widehat{\gamma}})$. This implies Eq.(55). (Q.E.D.)

References

[1] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, AC-19(6):716–723, 1974.

[2] A. Albert. *Regression and the Moore-Penrose Pseudoinverse.* Academic Press, New York and London, 1972.

[3] N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68:337–404, 1950.

[4] C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998.

[5] N. V. Chawla, N. Japkowicz, and A. Kotcz. Editorial: special issue on learning from imbalanced data sets. *ACM SIGKDD Explorations Newsletter*, 6(1):1–6, 2004.

[6] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1998.

[7] D. A. Cohn, Z. Ghahramani, and M. I. Jordan. Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4:129–145, 1996.

[8] T. Evgeniou, M. Pontil, and T. Poggio. Regularization networks and support vector machines. *Advances in Computational Mathematics*, 13(1):1–50, 2000.

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

[10] I. E. Frank and J. H. Friedman. A statistical view of some chemometrics regression tools. *Technometrics*, 35(2):109–135, 1993.

[11] K. Fukumizu. Statistical active learning in multilayer perceptrons. *IEEE Transactions on Neural Networks*, 11(1):17–26, 2000.

[12] W. Härdle, M. Müller, S. Sperlich, and A. Werwatz. *Nonparametric and Semiparametric Models.* Springer Series in Statistics. Springer, Berlin, 2004.

[13] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer, New York, 2001.

[14] J. J. Heckman. Sample selection bias as a specification error. *Econometrica*, 47(1):153–162, 1979.

[15] R. E. Henkel. *Tests of Significance.* SAGE Publication, Beverly Hills, 1979.

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

[17] M. Ishiguro, Y. Sakamoto, and G. Kitagawa. Bootstrapping log likelihood and EIC, an extension of AIC. *Annals of the Institute of Statistical Mathematics*, 49:411–434, 1997.

[18] T. Jaakkola and D. Haussler. Exploiting generative models in discriminative classiers. In M. Kearns, S.A. Solla, and D. Cohn, editors, *Advances in Neural Information Processing Systems 11*, pages 487–493. MIT Press, 1999.

[19] T. Kanamori and H. Shimodaira. Active learning algorithm using the maximum weighted log-likelihood estimator. *Journal of Statistical Planning and Inference*, 116(1):149–162, 2003.

[20] S. Konishi and G. Kitagawa. Generalized information criteria in model selection. *Biometrika*, 83:875–890, 1996.

[21] S. Lawrence, I. Burns, A. Back, A. C. Tsoi, and C. L. Giles. Neural network classification and prior class probabilities. In G. Orr and K.-R. Müller, editors, *Neural Networks: Tricks of the Trade*, Lecture Notes in Computer Science, pages 299–314. Springer-Verlag, Berlin, 1998.

[22] K. Li. Asymptotic optimality of C_L and generalized cross-validation in ridge regression with application to spline smoothing. *The Annals of Statistics*, 14(3):1101–1112, 1986.

[23] H. Linhart. A test whether two AIC's differ significantly. *South Africa Statistical Journal*, 22:153–161, 1988.

[24] A. Luntz and V. Brailovsky. On estimation of characters obtained in statistical procedure of recognition. *Technicheskaya Kibernetica*, 3, 1969. in Russian.

[25] D. J. C. MacKay. Information-based objective functions for active data selection. *Neural Computation*, 4(4):590–604, 1992.

[26] 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*, 5(6):865–872, 1994.

[27] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3):103–134, 2000.

[28] F. Pukelsheim. *Optimal Design of Experiments.* Wiley, 1993.

[29] C. R. Rao. *Linear Statistical Inference and Its Applications.* Wiley, New York, 1965.

[30] S. Saitoh. *Theory of Reproducing Kernels and Its Applications*, volume 189 of *Pitsman Research Notes in Mathematics Series.* Longman Scientific & Technical, UK, 1988.

[31] S. Saitoh. *Integral Transforms, Reproducing Kernels and Their Applications*, volume 369 of *Pitsman Research Notes in Mathematics Series.* Longman, UK, 1997.

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

[33] H. Shimodaira. Assessing the error probability of the model selection test. *Annals of Institute of Statistical Mathematics*, 49(3):395–410, 1997.

[34] H. Shimodaira. An application of multiple comparison techniques to model selection. *Annals of Institute of Statistical Mathematics*, 50(1):1–13, 1998.

[35] H. Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(2):227–244, 2000.

[36] B. W. Silverman. *Density Estimation for Statistics and Data Analysis.* Chapman and Hall, London, 1986.

[37] V. Spokoiny. Variance estimation for high-dimensional regression models. *Journal of Multivariate Analysis*, 82(1):111–133, 2002.

[38] M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society, Series B*, 36:111–147, 1974.

[39] M. Sugiyama, M. Kawanabe, and K.-R. Müller. Trading variance reduction with unbiasedness: The regularized subspace information criterion for robust model selection in kernel regression. *Neural Computation*, 16(5):1077–1104, 2004.

[40] M. Sugiyama and K.-R. Müller. The subspace information criterion for infinite dimensional hypothesis spaces. *Journal of Machine Learning Research*, 3(Nov):323–359, 2002.

[41] M. Sugiyama and H. Ogawa. Incremental active learning for optimal generalization. *Neural Computation*, 12(12):2909–2940, 2000.

[42] M. Sugiyama and H. Ogawa. Active learning for optimal generalization in trigonometric polynomial models. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E84-A(9):2319–2329, 2001.

[43] M. Sugiyama and H. Ogawa. Subspace information criterion for model selection. *Neural Computation*, 13(8):1863–1889, 2001.

[44] M. Sugiyama and H. Ogawa. Optimal design of regularization term and regularization parameter by subspace information criterion. *Neural Networks*, 15(3):349–361, 2002.

[45] J. A. K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle. *Least Squares Support Vector Machines.* World Scientific Pub. Co., Singapore, 2002.

[46] K. Takeuchi. Distribution of information statistics and validity criteria of models. *Mathematical Science*, 153:12–18, 1976. in Japanese.

[47] K. Takeuchi. On the selection of statistical models by AIC. *Journal of the Society of Instrument and Control Engineering*, 22(5):445–453, 1983. in Japanese.

[48] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58(1):267–288, 1996.

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

[50] G. Wahba. *Spline Model for Observational Data.* Society for Industrial and Applied Mathematics, Philadelphia and Pennsylvania, 1990.

[51] M. K. Warmuth, J. Liao, G. Rätsch, M. Mathieson, S. Putta, and C. Lemmen. Active learning with SVMs in the drug discovery process. *Chemical Information and Computer Sciences*, 43(2):667–673, 2003.

[52] P. M. Williams. Bayesian regularization and pruning using a Laplace prior. *Neural Computation*, 7(1):117–143, 1995.

[53] Z. Zhang. Parameter estimation techniques: A tutorial with application to conic fitting. *Image and Vision Computing*, 15(1):59–76, 1997.