

# Generalization Error Estimation under Covariate Shift

Masashi Sugiyama\*

Klaus-Robert Müller†

**Abstract:** In supervised learning, it is almost always assumed that the training and test input points follow the *same* probability distribution. However, this assumption is violated, e.g., in interpolation, extrapolation, active learning, or classification with imbalanced data. In such situations—known as the *covariate shift*, cross-validation estimate of the generalization error is biased, which results in poor model selection. In this paper, we propose an alternative estimator of the generalization error which is under the covariate shift exactly unbiased if model includes the learning target function and is 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 cross-validation.

**Keywords:** model selection, covariate shift, sample selection bias, extrapolation, active learning.

## 1 Introduction

We discuss a regression problem of learning  $f(\mathbf{x})$  from training examples

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

where  $\{\epsilon_i\}_{i=1}^n$  are i.i.d. noise with mean zero and unknown variance  $\sigma^2$ . We use the following linear regression model for learning<sup>1</sup>

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^p \alpha_i \varphi_i(\mathbf{x}), \quad (2)$$

where  $\{\varphi_i(\mathbf{x})\}_{i=1}^p$  are fixed linearly independent functions and  $p < n$ . We want to estimate the parameters  $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_p)^\top$  so that the expected square test error for all test input points (i.e., the *generalization error*) is minimized. When the test input points independently follow a distribution with density  $p_t(\mathbf{x})$ , the generalization error is expressed as

$$J = \int \left( \hat{f}(\mathbf{x}) - f(\mathbf{x}) \right)^2 p_t(\mathbf{x}) d\mathbf{x}. \quad (3)$$

It is most commonly assumed in supervised learning that the training input points  $\{\mathbf{x}_i\}_{i=1}^n$  are drawn

\*Department of Computer Science, Tokyo Institute of Technology, 2-12-1, O-okayama, Meguro-ku, Tokyo, 152-8552, Japan. TEL: 03-5734-2699 E-mail: sugi@cs.titech.ac.jp URL: <http://sugiyama-www.cs.titech.ac.jp/~sugi/>

†Fraunhofer FIRST.IDA, Kekuléstr. 7, 12489 Berlin, Germany, and Department of Computer Science, University of Potsdam, August-Bebel-Strasse 89, 14482 Potsdam, Germany. E-mail: klaus@first.fhg.de

<sup>1</sup>As detailed in the extended version [13], the present theory is still valid even in non-parametric scenarios where  $p$  increases as  $n$  increases.

independently from the *same* distribution as the test input points follow [15, 14, 4, 8]. 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 [2, 7]. 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 training and test input points follow different distributions is referred to as the situation under the *covariate shift* [9] or the *sample selection bias* [5], which we discuss in this paper. Let  $p_x(\mathbf{x})$  be the density of the training input points  $\{\mathbf{x}_i\}_{i=1}^n$ . An example of the extrapolation problem where  $p_x(\mathbf{x}) \neq p_t(\mathbf{x})$  is illustrated in Figure 1.

Under the covariate shift, two difficulties arise in a learning process. The first difficulty is parameter learning. The ordinary least-squares learning, given by

$$\min_{\boldsymbol{\alpha}} \left[ \sum_{i=1}^n \left( \hat{f}(\mathbf{x}_i) - y_i \right)^2 \right], \quad (4)$$

tries to fit the data well in the region with high training data density. This implies 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 true function is *unrealizable* (i.e., the learning target function is not included in the model at hand), least-squares learning is no longer *consistent* (i.e., the learned parameter does not converge to the optimal one even when the number of training

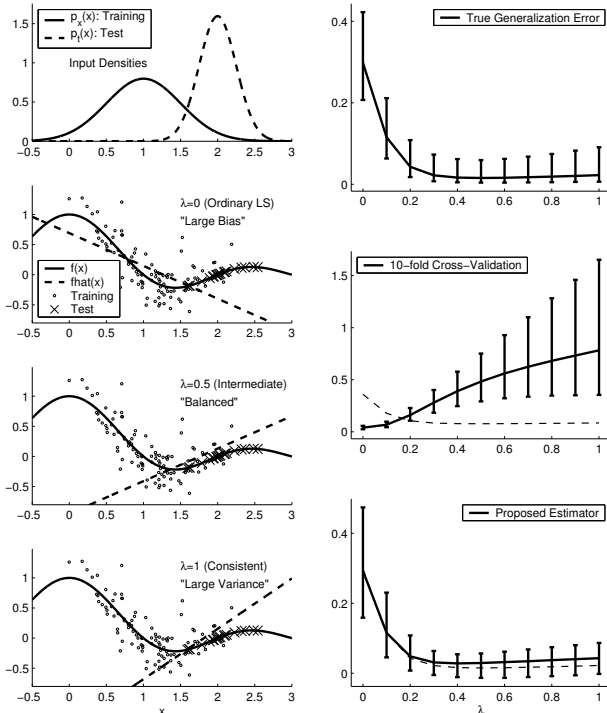


Figure 1: An illustrative example of extrapolation by fitting a linear function  $\hat{f}(\mathbf{x}) = \alpha_1 + \alpha_2 x$ . [Left column]: The top graph depicts the probability density functions of the training and test input points,  $p_x(x)$  and  $p_t(x)$ . In the bottom three graphs, the learning target function  $f(x)$  is drawn by the solid line, the noisy training examples are plotted with o's, a learned function  $\hat{f}(x)$  is drawn by the dashed line, and the (noiseless) test examples are plotted with x's. Three different learned functions are obtained by weighted least-squares learning with different tuning parameter  $\lambda$ .  $\lambda = 0$  corresponds to the ordinary least-squares learning (small variance but large bias), while  $\lambda = 1$  gives a consistent estimate (small bias but large variance). With finite samples, an intermediate  $\lambda$ , say  $\lambda = 0.5$ , often provides better results. [Right column]: The top graph depicts the mean and standard deviation of the generalization error over 300 independent trials, as a function of  $\lambda$ . The middle and bottom graphs depict the means and standard deviations of the estimated generalization error obtained by the standard 10-fold cross-validation (10CV) and the proposed method. The dotted lines are the mean of the true generalization error. 10CV is heavily biased because of  $p_x(x) \neq p_t(x)$ , while the proposed estimator is almost unbiased with reasonably small variance.

examples goes to infinity) under the covariate shift. This problem can be overcome by using a least-squares learning weighted by the *ratio* of test and training data

densities<sup>2</sup> [9].

$$\min_{\alpha} \left[ \sum_{i=1}^n \frac{p_t(\mathbf{x}_i)}{p_x(\mathbf{x}_i)} \left( \hat{f}(\mathbf{x}_i) - y_i \right)^2 \right]. \quad (5)$$

A key idea of this weighted version 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 least-squares learning tends to have large variance. Indeed, it is no longer *asymptotically efficient* even when the noise is Gaussian. Therefore, in practical situations with finite samples, a stabilized estimator, e.g.,

$$\min_{\alpha} \left[ \sum_{i=1}^n \left( \frac{p_t(\mathbf{x}_i)}{p_x(\mathbf{x}_i)} \right)^{\lambda} \left( \hat{f}(\mathbf{x}_i) - y_i \right)^2 \right], \quad (6)$$

where  $0 \leq \lambda \leq 1$ , would give more accurate estimates. The learned parameter  $\hat{\alpha}_{\lambda}$  obtained by the weighted least-squares learning (6) is given by

$$\hat{\alpha}_{\lambda} = \mathbf{L}_{\lambda} \mathbf{y}, \quad (7)$$

where

$$\mathbf{L}_{\lambda} = (\mathbf{X}^{\top} \mathbf{D}^{\lambda} \mathbf{X})^{-1} \mathbf{X}^{\top} \mathbf{D}^{\lambda}, \quad (8)$$

$$\mathbf{X}_{i,j} = \varphi_j(\mathbf{x}_i), \quad (9)$$

$$\mathbf{D} = \text{diag} \left( \frac{p_t(\mathbf{x}_1)}{p_x(\mathbf{x}_1)}, \frac{p_t(\mathbf{x}_2)}{p_x(\mathbf{x}_2)}, \dots, \frac{p_t(\mathbf{x}_n)}{p_x(\mathbf{x}_n)} \right), \quad (10)$$

$$\mathbf{y} = (y_1, y_2, \dots, y_n)^{\top}. \quad (11)$$

Note that  $\lambda = 0$  corresponds to the ordinary least-squares learning (4), while  $\lambda = 1$  corresponds to the consistent weighted least-squares learning (5). Thus, the parameter learning problem is now relocated to the model selection problem of choosing  $\lambda$ .

However, the second difficulty when  $p_x(\mathbf{x}) \neq p_t(\mathbf{x})$  is model selection itself. Standard unbiased generalization error estimation schemes such as cross-validation [6, 11, 15] 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.

In this paper, we therefore propose a *new* generalization error estimator. Under the covariate shift, the proposed estimator is proved to be exactly unbiased with finite samples in realizable cases and asymptotically unbiased in general. Furthermore, the proposed generalization error estimator is shown to be able to accurately estimate the *difference* of the generalization error, which is a useful property in model selection.

For simplicity, we focus on the problem of choosing the tuning parameter  $\lambda$  (see Eq.(6)) in the following. However, the proposed theory can be easily extended to general model selection of choosing basis functions or regularization constant (see [13] for detail).

<sup>2</sup>For the moment, we assume that  $p_x(\mathbf{x})$  and  $p_t(\mathbf{x})$  are known. Later we theoretically and experimentally investigate the cases where they are unknown and estimated from data.

## 2 Derivation of Generalization Error Estimator

Let us decompose the learning target function  $f(\mathbf{x})$  into

$$f(\mathbf{x}) = g(\mathbf{x}) + r(\mathbf{x}), \quad (12)$$

where  $g(\mathbf{x})$  is the orthogonal projection of  $f(\mathbf{x})$  onto the span of  $\{\varphi_i(\mathbf{x})\}_{i=1}^p$  and the residual  $r(\mathbf{x})$  is orthogonal to  $\{\varphi_i(\mathbf{x})\}_{i=1}^p$ , i.e., for  $i = 1, 2, \dots, p$ ,

$$\int_{\mathcal{D}} r(\mathbf{x}) \varphi_i(\mathbf{x}) p_t(\mathbf{x}) d\mathbf{x} = 0. \quad (13)$$

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

$$g(\mathbf{x}) = \sum_{i=1}^p \alpha_i^* \varphi_i(\mathbf{x}), \quad (14)$$

where  $\boldsymbol{\alpha}^* = (\alpha_1^*, \alpha_2^*, \dots, \alpha_p^*)^\top$  are unknown optimal parameters.

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

$$U_{i,j} = \int_{\mathcal{D}} \varphi_i(\mathbf{x}) \varphi_j(\mathbf{x}) p_t(\mathbf{x}) d\mathbf{x}, \quad (15)$$

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

$$\begin{aligned} J(\lambda) &= \int \widehat{f}_\lambda(\mathbf{x})^2 p_t(\mathbf{x}) d\mathbf{x} - 2 \int \widehat{f}_\lambda(\mathbf{x}) f(\mathbf{x}) p_t(\mathbf{x}) d\mathbf{x} \\ &\quad + \int f(\mathbf{x})^2 p_t(\mathbf{x}) d\mathbf{x} \\ &= \langle \mathbf{U} \widehat{\boldsymbol{\alpha}}_\lambda, \widehat{\boldsymbol{\alpha}}_\lambda \rangle - 2 \langle \mathbf{U} \widehat{\boldsymbol{\alpha}}_\lambda, \boldsymbol{\alpha}^* \rangle + C, \end{aligned} \quad (16)$$

where

$$C = \int_{\mathcal{D}} f(\mathbf{x})^2 p_t(\mathbf{x}) d\mathbf{x}. \quad (17)$$

In Eq.(16), the first term  $\langle \mathbf{U} \widehat{\boldsymbol{\alpha}}_\lambda, \widehat{\boldsymbol{\alpha}}_\lambda \rangle$  is accessible and the third term  $C$  does not depend on  $\lambda$ . Therefore, we focus on estimating the second term “ $-2 \langle \mathbf{U} \widehat{\boldsymbol{\alpha}}_\lambda, \boldsymbol{\alpha}^* \rangle$ ”.

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

$$\mathbb{E}_\epsilon \mathbf{L}_u \mathbf{y} = \boldsymbol{\alpha}^*, \quad (18)$$

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

$$\begin{aligned} \mathbb{E}_\epsilon \langle \mathbf{U} \widehat{\boldsymbol{\alpha}}_\lambda, \boldsymbol{\alpha}^* \rangle &= \langle \mathbb{E}_\epsilon \mathbf{U} \mathbf{L}_\lambda \mathbf{y}, \mathbb{E}_\epsilon \mathbf{L}_u \mathbf{y} \rangle \\ &= \mathbb{E}_\epsilon \langle \mathbf{U} \mathbf{L}_\lambda \mathbf{y}, \mathbf{L}_u \mathbf{y} \rangle \\ &\quad - \sigma^2 \text{tr}(\mathbf{U} \mathbf{L}_\lambda \mathbf{L}_u^\top). \end{aligned} \quad (19)$$

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

$$\begin{aligned} \mathbb{E}_\epsilon [\langle \mathbf{U} \mathbf{L}_\lambda \mathbf{y}, \mathbf{L}_u \mathbf{y} \rangle - \sigma_u^2 \text{tr}(\mathbf{U} \mathbf{L}_\lambda \mathbf{L}_u^\top)] \\ = \mathbb{E}_\epsilon \langle \mathbf{U} \widehat{\boldsymbol{\alpha}}_\lambda, \boldsymbol{\alpha}^* \rangle. \end{aligned} \quad (20)$$

However, either  $\mathbf{L}_u$  or  $\sigma_u^2$  could be unavailable<sup>3</sup>. So we use the following approximations instead:

$$\widehat{\mathbf{L}}_u = (\mathbf{X}^\top \mathbf{D} \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{D}, \quad (21)$$

$$\widehat{\sigma}_u^2 = \|\mathbf{G} \mathbf{y}\|^2 / \text{tr}(\mathbf{G}), \quad (22)$$

where

$$\mathbf{G} = \mathbf{I} - \mathbf{X}(\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top, \quad (23)$$

and  $\mathbf{I}$  denotes the identity matrix.

Eq.(21) is actually the learning matrix corresponding to the consistent weighted least-squares learning described in Eq.(5), implying that it exactly fulfills Eq.(18) in realizable cases and it asymptotically satisfies Eq.(18) in general [9].

On the other hand, it is known that  $\widehat{\sigma}_u^2$  is an exact unbiased estimator of  $\sigma^2$  in realizable cases [2]. In unrealizable cases, however, it is not unbiased even asymptotically. Although it is possible to obtain asymptotic unbiased estimators of  $\sigma^2$  under some smoothness assumption on  $f(\mathbf{x})$  [10], 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  $J$ .

$$\begin{aligned} \widehat{J}(\lambda) &= \langle \mathbf{U} \mathbf{L}_\lambda \mathbf{y}, \mathbf{L}_\lambda \mathbf{y} \rangle - 2 \langle \mathbf{U} \mathbf{L}_\lambda \mathbf{y}, \widehat{\mathbf{L}}_u \mathbf{y} \rangle \\ &\quad + 2 \widehat{\sigma}_u^2 \text{tr}(\mathbf{U} \mathbf{L}_\lambda \widehat{\mathbf{L}}_u^\top). \end{aligned} \quad (24)$$

## 3 Theoretical Analyses

In this section, theoretical properties of the proposed generalization error estimator  $\widehat{J}$  is investigated.

### 3.1 Unbiasedness

Let  $B_\epsilon$  be the bias of  $\widehat{J}$ :

$$B_\epsilon = \mathbb{E}_\epsilon [\widehat{J} - J] + C, \quad (25)$$

Then we have the following theorem (Proofs of all theorems and lemmas are available in [13]).

**Theorem 1** *If  $r(\mathbf{x}_i) = 0$  for  $i = 1, 2, \dots, n$ ,*

$$B_\epsilon = 0. \quad (26)$$

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

$$B_\epsilon = \mathcal{O}(\delta). \quad (27)$$

*If  $n$  is sufficiently large,*

$$B_\epsilon = \mathcal{O}_p(n^{-\frac{1}{2}}). \quad (28)$$

<sup>3</sup>Note that  $\mathbf{L}_u$  is always available if the functional Hilbert space  $\mathcal{H}$  has the reproducing kernel and the span of the basis functions  $\{\varphi_i(\mathbf{x})\}_{i=1}^p$  is included in the span of  $\{K(\mathbf{x}, \mathbf{x}_i)\}_{i=1}^n$  [12], where  $K(\mathbf{x}, \mathbf{x}')$  is the reproducing kernel. In this paper, however, we consider general functional Hilbert spaces and general basis functions which may not satisfy such conditions.

Note that the asymptotic order in Eq.(28) is in probability because the expectation over  $\{\mathbf{x}_i\}_{i=1}^n$  is not taken. This theorem implies that, except for the constant  $C$ ,  $\hat{J}$  is exactly unbiased if  $f(\mathbf{x})$  is strictly realizable, it is almost unbiased if  $f(\mathbf{x})$  is almost realizable, and it is asymptotically unbiased in general.

### 3.2 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, we want to accurately estimate the *difference* of the generalization error among different models. Here, we show that the proposed generalization error estimator  $\hat{J}$  is useful for this purpose. Recall that 'model' in the current setting<sup>4</sup> refers to  $\lambda$ .

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

$$\Delta B_\epsilon = \mathbb{E}_\epsilon[\Delta\hat{J} - \Delta J]. \quad (29)$$

If the "size" of  $\Delta B_\epsilon$  is smaller than that of  $\mathbb{E}_\epsilon[\Delta J]$ , then  $\hat{J}$  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  $\hat{J}$  is *effective in model comparison for  $\mathcal{M}$*  if

$$|\Delta B_\epsilon| < |\mathbb{E}_\epsilon[\Delta J]| \quad (30)$$

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

$$\Delta B_\epsilon = \mathcal{O}_p(n^{-s}) \quad \text{and} \quad \mathbb{E}_\epsilon[\Delta J] = \mathcal{O}_p(n^{-t}) \quad (31)$$

with  $s > t$ . First, the following corollary holds.

**Corollary 2** *If  $r(\mathbf{x}_i) = 0$  for  $i = 1, 2, \dots, n$ , then for any two different  $\lambda$*

$$\Delta B_\epsilon = 0. \quad (32)$$

This implies that if  $f(\mathbf{x})$  is realizable and  $|\mathbb{E}_\epsilon[\Delta J]| > 0$ ,  $\hat{J}$  is effective in model comparison. Similarly, we have

$$\Delta B_\epsilon = \mathcal{O}(\delta), \quad (33)$$

implying that  $\hat{J}$  is useful for model comparison if  $f(\mathbf{x})$  is almost realizable. In general cases where  $f(\mathbf{x})$  is not realizable, we have the following corollary.

**Corollary 3** *If two learned functions obtained from two different models converge to different functions,*

$$\Delta B_\epsilon = \mathcal{O}_p(n^{-\frac{1}{2}}) \quad \text{and} \quad \mathbb{E}_\epsilon[\Delta J] = \mathcal{O}_p(1). \quad (34)$$

If  $\lambda$  is different, learned functions generally converge to different functions. Therefore, in comparison of such models,  $\hat{J}$  is asymptotically effective.

<sup>4</sup>In the extended version [13], more general analyses are given.

### 3.3 When $p_x(\mathbf{x})$ and $p_t(\mathbf{x})$ Are Unknown

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

$p_t(\mathbf{x})$  is contained in  $\mathbf{U}$  and  $\hat{\mathbf{L}}_u$ , while  $p_x(\mathbf{x})$  appears only in  $\hat{\mathbf{L}}_u$ . So we investigate the effect of replacing  $p_x(\mathbf{x})$  and  $p_t(\mathbf{x})$  included in  $\mathbf{U}$  and  $\hat{\mathbf{L}}_u$  with their estimates. Note that  $p_x(\mathbf{x})$  and  $p_t(\mathbf{x})$  also appear in the learning matrix  $\mathbf{L}_\lambda$  via  $\mathbf{D}$ , which should also be replaced by the estimates. However, we here aim to investigate the accuracy of  $\hat{J}$  as a function of  $\mathbf{L}_\lambda$ , so  $p_x(\mathbf{x})$  and  $p_t(\mathbf{x})$  included in  $\mathbf{L}_\lambda$  do not have to be taken into account.

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

**Lemma 4** *Let*

$$\eta_t = \max\{|\hat{p}_t(\mathbf{x}_i) - p_t(\mathbf{x}_i)|\}_{i=1}^n, \quad (35)$$

$$\xi_t = \max\left\{\left|\int_{\mathcal{D}} \varphi_i(\mathbf{x})\varphi_j(\mathbf{x})(\hat{p}_t(\mathbf{x}) - p_t(\mathbf{x}))d\mathbf{x}\right|\right\}_{i,j=1}^n. \quad (36)$$

*If  $\eta_t$  and  $\xi_t$  are sufficiently small,*

$$\hat{J}_t = \hat{J} + \mathcal{O}(\eta_t + \xi_t). \quad (37)$$

This lemma states that if a reasonably good estimator  $\hat{p}_t(\mathbf{x})$  of the true density function  $p_t(\mathbf{x})$  is available, a good approximation of  $\hat{J}$  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  $p_t(\mathbf{x})$ . Actually, in some application domains—e.g., document classification or bioinformatics—a large number of unlabeled samples are easily gathered. In such cases, a reasonably good estimator  $\hat{p}_t(\mathbf{x})$  may be obtained by some standard density estimation methods.

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

**Lemma 5** *Let*

$$\eta_x = \max\{|\hat{p}_x(\mathbf{x}_i) - p_x(\mathbf{x}_i)|\}_{i=1}^n, \quad (38)$$

$$\gamma = \min\{p_x(\mathbf{x}_i)\}_{i=1}^n, \quad (39)$$

$$\hat{\gamma} = \min\{\hat{p}_x(\mathbf{x}_i)\}_{i=1}^n. \quad (40)$$

*If  $\gamma > 0$  and  $\hat{\gamma} > 0$ , and if  $\eta_x$  is sufficiently small,*

$$\hat{J}_x = \hat{J} + \mathcal{O}\left(\frac{\eta_x}{\gamma\hat{\gamma}}\right). \quad (41)$$

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

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

## 4 Numerical Examples

Figure 1 shows the numerical results of an illustrative extrapolation problem. The curves in the right column show that the proposed estimator gives almost unbiased estimates of the generalization error with reasonably small variance (note that the target function is not realizable in this case).

We also applied the proposed method to *Abalone* data set available from the UCI repository [1]. 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 of the 4-th input variable (weight of abalones) has negative bias 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. We used multi-dimensional linear basis functions for learning. Here we suppose that the test input points are known (i.e., the setting corresponds to transductive inference [14]) and the density functions  $p_x(\mathbf{x})$  and  $p_t(\mathbf{x})$  were estimated from the training input points and test input points, respectively, using a standard kernel density estimation method with the Gaussian kernel and *Silverman’s rule-of-thumb bandwidth selection rule* [3].

Figure 2 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 (due to the resampling process) and deteriorated the graphs. Note that the true generalization error  $J$  is calculated using the test examples. The proposed  $\hat{J}$  seems to give reasonably good curves and its minimum roughly agrees with the minimum of the true test error. On the other hand, irrespective of  $n$ , the minimizing value of the tuning parameter  $\lambda$  estimated by 10CV tends to be small.

We chose the tuning parameter  $\lambda$  by each method and estimated the age of the test abalones by using

Table 1: Extrapolation of the 4-th variable (top) or the 6-th variable (bottom) in the Abalone dataset. The mean and standard deviation of the test error obtained with each method are described. The better method and comparable one by the *t-test* at the significance level 5% are described with boldface.

$n$	$\hat{J}$	10CV
50	<b>11.67 ± 5.74</b>	<b>10.88 ± 5.05</b>
200	<b>7.95 ± 2.15</b>	<b>8.06 ± 1.91</b>
800	<b>6.77 ± 1.40</b>	7.23 ± 1.37

$n$	$\hat{J}$	10CV
50	<b>10.67 ± 6.19</b>	<b>10.15 ± 4.95</b>
200	<b>7.31 ± 2.24</b>	<b>7.42 ± 1.81</b>
800	<b>6.20 ± 1.33</b>	6.68 ± 1.25

the chosen  $\lambda$ . 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 are described in the top of Table 1. It shows that  $\hat{J}$  and 10CV work comparably for  $n = 50, 200$ , while  $\hat{J}$  outperforms 10CV for  $n = 800$ . Hence, the proposed method overall compares favorably to 10CV.

We also carried out similar simulations when the sampling of the 6-th input variable (weight of gut after bleeding) is biased. The results described in the bottom of Table 1 showed similar trends to the previous ones.

Application to imbalanced classification is available in the extended version [13], where the proposed method also showed promising results.

## 5 Conclusions

We proposed a new generalization error estimator under the covariate shift paradigm. The proposed estimator is proved to be unbiased with finite samples in realizable cases and asymptotically unbiased in general. We also showed that it is effective in model comparison. Although in theory, we assumed that the density functions of training and test input points are both known, Lemma 4 and Lemma 5 guarantee that the proposed method is still valid as long as reasonable density estimates particularly at each data point are available. Competitive results to cross-validation in experiments on toy and benchmark data underline our theoretical findings. Future applications of the current framework will consider data from brain computer interfacing experiments, where the statistics is known to change from the training session to the feedback session.

**Acknowledgments:** The authors would like to thank Dr. Motoaki Kawanabe and Dr. Gilles Blan-

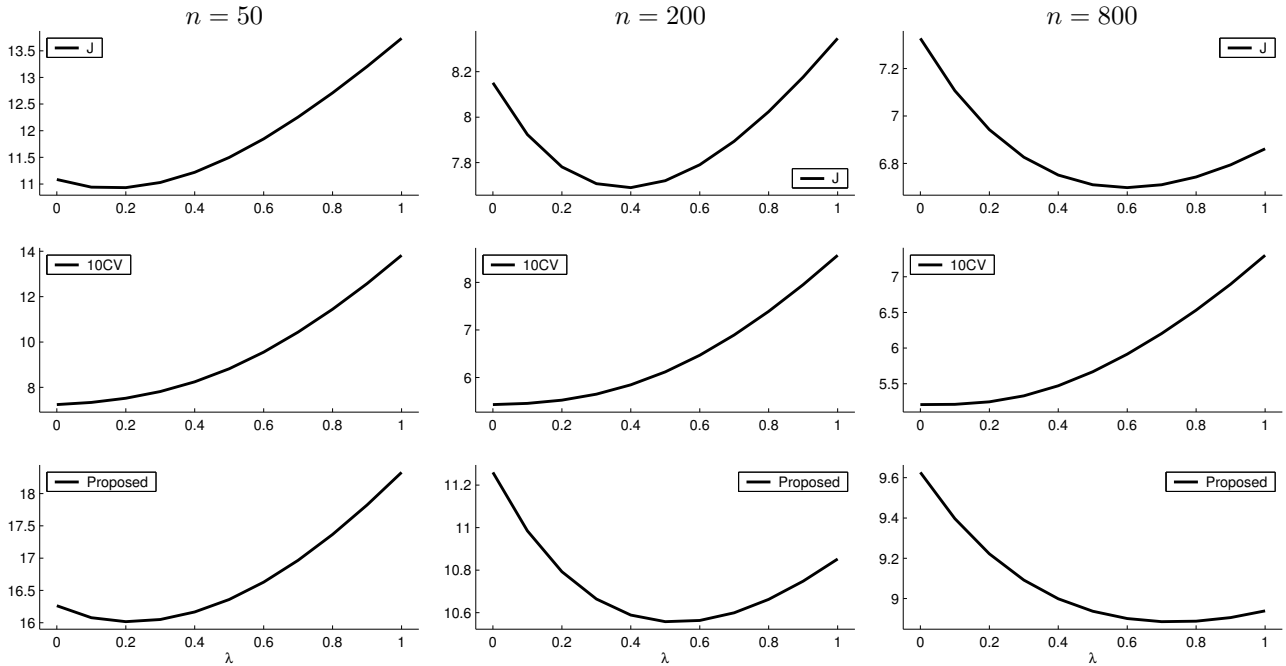


Figure 2: Extrapolation of the 4-th variable in the Abalone dataset. The mean of each method is described. Each column corresponds to each  $n$ .

chard for their valuable comments. Most of the work has been carried out when M.S. stayed at Fraunhofer FIRST, Berlin, Germany, which is supported by the Alexander von Humboldt Foundation. We acknowledge MEXT (Grant-in-Aid for Young Scientists 17700142) and the PASCAL Network of Excellence (EU #506778) for financial support.

## References

- [1] C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998.
- [2] V. V. Fedorov. *Theory of Optimal Experiments*. Academic Press, New York, 1972.
- [3] W. Härdle, M. Müller, S. Sperlich, and A. Werwatz. *Nonparametric and Semiparametric Models*. Springer Series in Statistics. Springer, Berlin, 2004.
- [4] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, New York, 2001.
- [5] J. J. Heckman. Sample selection bias as a specification error. *Econometrica*, 47(1):153–162, 1979.
- [6] A. Luntz and V. Brailovsky. On estimation of characters obtained in statistical procedure of recognition. *Technicheskaya Kibernetika*, 3, 1969. in Russian.
- [7] F. Pukelsheim. *Optimal Design of Experiments*. John Wiley & Sons, 1993.
- [8] B. Schölkopf and A. J. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.
- [9] 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.
- [10] V. Spokoiny. Variance estimation for high-dimensional regression models. *Journal of Multivariate Analysis*, 82(1):111–133, 2002.
- [11] M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society, Series B*, 36:111–147, 1974.
- [12] 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.
- [13] M. Sugiyama and K.-R. Müller. Model selection when training and test input points follow different distributions. Technical Report TR05-0001, Department of Computer Science, Tokyo Institute of Technology, May 2005.
- [14] V. N. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, Inc., New York, 1998.
- [15] G. Wahba. *Spline Model for Observational Data*. Society for Industrial and Applied Mathematics, Philadelphia and Pennsylvania, 1990.