Journal of Machine Learning Research 7 (2006) 141–166 Submitted 6/05; Revised 9/05 and 12/05; Published 1/06

Active Learning in Approximately Linear Regression Based on Conditional Expectation of Generalization Error

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

Editor: Greg Ridgeway

Abstract

The goal of active learning is to determine the locations of training input points so that the generalization error is minimized. We discuss the problem of active learning in linear regression scenarios. Traditional active learning methods using least-squares learning often assume that the model used for learning is correctly specified. In many practical situations, however, this assumption may not be fulfilled. Recently, active learning methods using "importance"-weighted least-squares learning have been proposed, which are shown to be robust against misspecification of models. In this paper, we propose a new active learning method also using the weighted least-squares learning, which we call *ALICE* (Active Learning using the Importance-weighted least-squares learning based on Conditional Expectation of the generalization error). An important difference from existing methods is that we predict the *conditional* expectation of the generalization error given training input points, while existing methods predict the *full* expectation of the generalization error. Due to this difference, the training input design can be fine-tuned depending on the realization of training input points. Theoretically, we prove that the proposed active learning criterion is a more accurate predictor of the *single-trial* generalization error than the existing criterion. Numerical studies with toy and benchmark data sets show that the proposed method compares favorably to existing methods.

Keywords: Active Learning, Conditional Expectation of Generalization Error, Misspecification of Models, Importance-Weighted Least-Squares Learning, Covariate Shift.

1. Introduction

In a standard setting of supervised learning, the training input points are provided from the environment (Vapnik, 1998). On the other hand, there are cases where the location of the training input points can be designed by users (Fedorov, 1972; Pukelsheim, 1993). In such situations, it is expected that the accuracy of learned results can be improved by appropriately choosing the location of the training input points, e.g., by densely allocating the training input points in the regions with high uncertainty. *Active learning* (MacKay, 1992; Cohn et al., 1996; Fukumizu, 2000)—also referred to as *experimental design* in statistics (Kiefer, 1959; Fedorov, 1972; Pukelsheim, 1993)—is the problem of optimizing location of training input points so that the generalization error is minimized.

The generalization error can be decomposed into the *bias* and *variance* terms. In active learning research, it is often assumed that the model used for learning is correctly specified (Fedorov, 1972; Cohn et al., 1996; Fukumizu, 2000), i.e., the learning target function can be expressed by the model. Then, under a mild condition, the ordinary least-squares (OLS) learning yields that the bias term vanishes and only the variance term remains. Based on this fact, a traditional active learning method

with OLS tries to determine the location of the training input points so that the variance term is minimized (Fedorov, 1972). In practice, however, the correctness of the model may not be fulfilled.

Active learning is a situation under the *covariate shift* (Shimodaira, 2000), where the training input distribution is different from the test input distribution. When the model used for learning is correctly specified, the covariate shift does not matter because OLS is still unbiased under a mild condition. However, OLS is no longer unbiased even asymptotically for misspecified models, and therefore we have to explicitly deal with the bias term if OLS is used.

Under the covariate shift, it is known that a form of weighted least-squares learning (WLS) is shown to be asymptotically unbiased even for misspecified models (Shimodaira, 2000; Wiens, 2000). The key idea of this WLS is the use of the ratio of density functions of test and training input points: the goodness-of-fit of the training input points is adjusted to that of the test input points by the density ratio, which is similar to *importance sampling*.

In this paper, we propose a variance-only active learning method using WLS, which can be regarded as an extension of the traditional variance-only active learning method using OLS. The proposed method can be theoretically justified for the approximately correct models, and thus is *robust* against the misspecification of models.

Conditional Expectation of Generalization Error: A variance-only active learning method using WLS has also been proposed by Wiens (2000), which can also be theoretically justified for approximately correct models. The important difference is how the generalization error is predicted: we predict the *conditional* expectation of the generalization error given training input points, while in Wiens (2000), the *full* expectation of the generalization error is predicted. In order to explain this difference in more detail, we first note that the generalization error of the WLS estimator depends on the training input density since WLS explicitly uses it. Therefore, when WLS is used in active learning, the generalization error is predicted as a function of the training input density, and the training input density is optimized so that the predicted generalization error is minimized.

The parameters in the model are learned using the training examples, which consist of training input points drawn from the user-designed distribution and corresponding noisy output values. This means that the generalization error is a random variable which depends on the location of the training input points and noise contained in the training output values. We ideally want to predict the *single-trial* generalization error, i.e., the generalization error for a single realization of the training examples at hand. From this viewpoint, we do not want to average out the random variables, but we want to plug the realization of the random variables into the generalization error and evaluate the realized value of the generalization error. However, we may not be able to avoid taking the expectation over the training output noise since the training output noise is inaccessible. In contrast, the location of the training input points are accessible by nature. Motivated by this fact, in this paper, we predict the generalization error *without* taking the expectation over the training input points. That is, we predict the *conditional* expectation of the generalization error given training input points. On the other hand, in Wiens (2000), the generalization error is predicted in terms of the expectation over *both* the training input points and the training output noise.

A possible advantage of the conditional-expectation approach is schematically illustrated in Figure 1. For illustration purposes, we consider the case of sampling only one training example. The solid curves in the left graph (Figure 1-(a)) depict $G_{p_a}(\varepsilon|x)$, the generalization error for a training input density p_a as a function of the training output noise ε given a training input point x. The three solid curves correspond to the cases where the realizations of the training input point x are a_1, a_2,

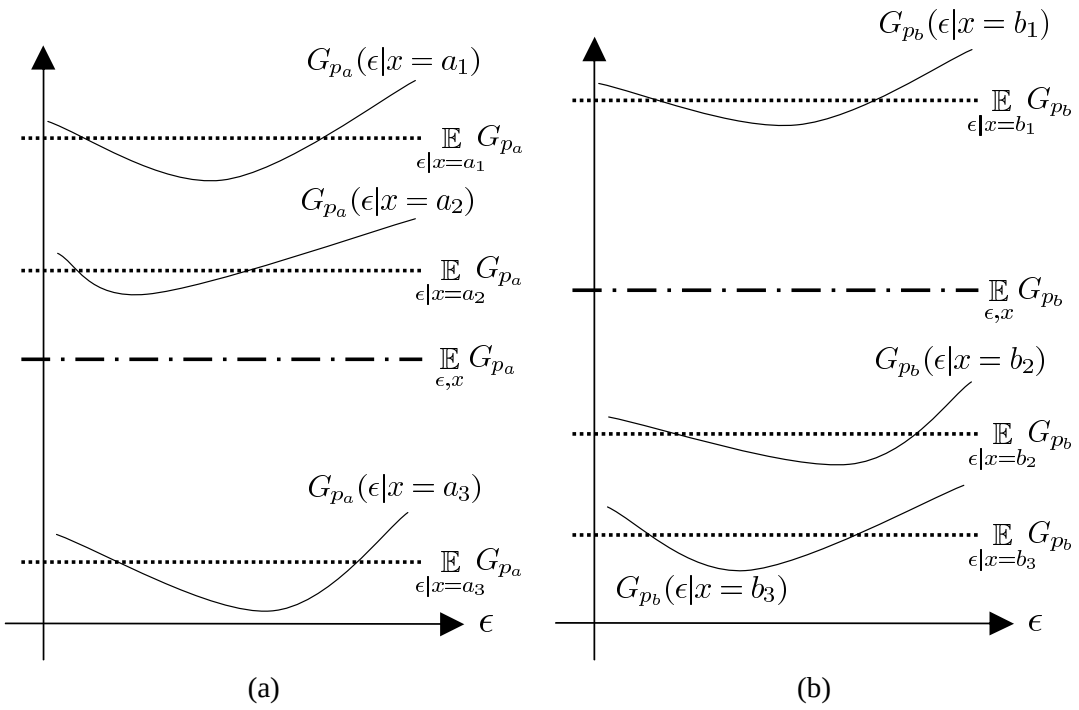

Figure 1: Schematic illustration of conditional expectation and full expectation of the generalization error. (a) and (b) correspond to the generalization error for p_a and p_b, respectively.

and a_3, respectively. The value of the generalization error for the density p_a in the full-expectation approach is depicted by the dash-dotted line, where the generalization error is expected over both the training output noise ε and the training input points x (i.e., the mean of the three solid curves). The values of the generalization error in the conditional-expectation approach are depicted by the dotted lines, where the generalization errors are expected only over the training output noise ε, given $x = a_1, a_2, a_3$, respectively (i.e., the mean of each solid curve). The right graph (Figure 1-(b)) depicts the generalization errors for the training input density p_b in the same manner.

In the full-expectation framework, the density p_a is judged to be better than p_b regardless of the realization of the training input point since the dash-dotted line in the left graph is lower than that in the right graph (see Figure 1 again). However, as the solid curves show, p_a is often worse than p_b in single trials. On the other hand, in the conditional-expectation framework, the goodness of the density is adaptively judged depending on the realizations of the training input point x. For example, p_b is judged to be better than p_a if a_2 and b_3 are realized, or p_a is judged to be better than p_b if a_3 and b_1 are realized. That is, the conditional-expectation framework may yield a better choice of the training input density (and the training input points) than the full-expectation framework.

The above discussion illustrates a conceptual advantage of the conditional-expectation approach. Theoretically, we prove that the proposed active learning criterion derived in the conditional-expectation framework is a better predictor of the single-trial generalization error than the full-expectation active learning criterion proposed by Wiens (2000). This substantiates the advantage of the conditional-expectation approach. Experimental results also support this claim: the

proposed method compares favorably to Wiens's method in the simulations with toy and benchmark data sets.

Bias-and-Variance Approach for Misspecified Models: Kanamori and Shimodaira (2003) also proposed an active learning algorithm using WLS. This method is not variance-only, but it takes both the bias and the variance into account by gathering training input points in two stages. In the first stage, a certain number of training examples are randomly gathered from the environment, and the generalization error (i.e., the sum of the bias and variance) is predicted by using the gathered training examples. Then in the second stage, the training input density for the remaining training examples is optimized based on the generalization error prediction. Theoretically, the two-stage method is shown to asymptotically give the optimal training input density not only for approximately correct models, but also for totally misspecified models. Although this property is solid, it may not be practically valuable since learning with totally misspecified models may not work well because of the model error. A drawback of this method is that it requires some randomly collected training examples in the first stage, so we are not allowed to optimally design all the training input locations by ourselves. Our experiments show that the proposed method works better than the two-stage method of Kanamori and Shimodaira (2003).

Batch Selection of Training Input Points: Active learning in the machine learning community is often thought of as being a *sequential* process: selecting one or a few training input points, observing corresponding training output values, training the model using the gathered training examples, and iterating this process. An alternative approach is the *batch* approach, where all training input points are gathered in the beginning.

If the environment is non-stationary, i.e., the learning target function drifts, taking the sequential approach would be necessary. On the other hand, under the stationary environment, i.e., the learning target function is fixed, the batch approach gives the globally optimal solution and the sequential approach can be regarded as a greedy approximation to it. In this paper, we consider the stationary case, so the batch approach is desirable.

In correctly specified linear regression, the expected generalization error does not depend on the learning target function under a mild condition. Therefore, the globally optimal solution can be obtained in principle. However, in misspecified linear regression which we discuss in this paper, the expected generalization error depends on the unknown learning target function. In this scenario, the sequential approach would be natural: estimating the unknown learning target function and optimizing location of the training input points are carried out alternately. On the other hand, in this paper, we do not estimate the learning target function, but we approximate the generalization error by the quantity which does *not* depend on the learning target function. This makes it possible to take the batch approach of determining all the training input points at once in advance.

A general criticism of the batch approach is that except for some special cases where the global optimal solution can be obtained analytically (Fedorov, 1972; Sugiyama and Ogawa, 2001), the batch approach usually requires the simultaneous optimization of all training input points, which is computationally very demanding. On the other hand, the sequential approach is computationally efficient since only one or a few training input points are optimized in each iteration (Cohn et al., 1996; Fukumizu, 2000; Sugiyama and Ogawa, 2000). In this paper, we avoid the computational difficulty of the batch approach not by resorting to the sequential approach, but by optimizing the training input distribution, rather than directly optimizing the training input points themselves. This

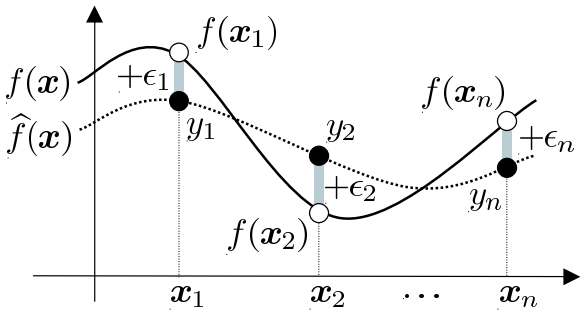

Figure 2: Regression problem.

seems to be a popular approach in batch active learning research (Wiens, 2000; Kanamori and Shimodaira, 2003).

Organization: The rest of this paper is organized as follows. We derive a new active learning method in Section 2, and we discuss relations between the proposed method and the existing methods in Section 3. We report numerical results using toy and benchmark data sets in Section 4. Finally, we state conclusions and future prospects in Section 5.

2. Derivation of New Active Learning Method

In this section, we formulate the active learning problem in regression scenarios, and derive a new active learning method.

2.1 Problem Formulation

Let us discuss the regression problem of learning a real-valued function $f(x)$ defined on $\mathbb{R}^d$ from training examples (see Figure 2). Training examples are given as

$$\{(x_i, y_i) \mid y_i = f(x_i) + \varepsilon_i\}_{i=1}^n,$$

where $\{\varepsilon_i\}_{i=1}^n$ are i.i.d. noise with mean zero and unknown variance σ^2. We suppose that the training input points $\{x_i\}_{i=1}^n$ are independently drawn from a user-defined distribution with density $p(x)$.

Let $\widehat{f}(x)$ be a learned function obtained from the training examples $\{(x_i, y_i)\}_{i=1}^n$. We evaluate the goodness of the learned function $\widehat{f}(x)$ by the expected squared test error over test input points, to which refer as the *generalization error*. When the test input points are drawn independently from a distribution with density $q(x)$, the generalization error G' is expressed as

$$G' = \int \left(\widehat{f}(x) - f(x)\right)^2 q(x) dx. \tag{1}$$

We suppose that $q(x)$ is known (or its reasonable estimate is available). This seems to be a common assumption in active learning literature (e.g., Fukumizu, 2000; Wiens, 2000; Kanamori and Shimodaira, 2003). If a large number of *unlabeled samples*[1] are easily gathered, a reasonably good

1. Unlabeled samples are input points without output values. We assume that unlabeled samples are independently drawn from the distribution with density $q(x)$.

estimate of $q(x)$ may be obtained by some standard density estimation method. Therefore, the assumption that $q(x)$ is known or its reasonable estimate is available may not be so restrictive.

In the following, we discuss the problem of optimizing the training input density $p(x)$ so that the generalization error is minimized.

2.2 Approximately Correct Linear Regression

We learn the target function $f(x)$ by the following linear regression model:

$$\widehat{f}(x) = \sum_{i=1}^{b} \widehat{\alpha}_i \varphi_i(x), \tag{2}$$

where $\{\varphi_i(x)\}_{i=1}^{b}$ are fixed linearly independent functions[2] and $\widehat{\alpha} = (\widehat{\alpha}_1, \widehat{\alpha}_2, \ldots, \widehat{\alpha}_b)^\top$ are parameters to be learned (by a variant of least-squares, see Section 2.4 for detail).

Suppose the regression model (2) does not exactly include the learning target function $f(x)$, but it *approximately* includes it, i.e., for a scalar δ such that $|\delta|$ is small, $f(x)$ is expressed as

$$f(x) = g(x) + \delta r(x), \tag{3}$$

where $g(x)$ is the optimal approximation to $f(x)$ by the model (2):

$$g(x) = \sum_{i=1}^{b} \alpha_i^* \varphi_i(x).$$

$\alpha^* = (\alpha_1^*, \alpha_2^*, \ldots, \alpha_b^*)^\top$ is the unknown optimal parameter defined by

$$\alpha^* = \operatorname*{argmin}_{\alpha} \int \left(\sum_{i=1}^{b} \alpha_i \varphi_i(x) - f(x) \right)^2 q(x) dx.$$

$\delta r(x)$ in Eq.(3) is the residual, which is orthogonal to $\{\varphi_i(x)\}_{i=1}^{b}$ under $q(x)$ (see Figure 3):

$$\int r(x) \varphi_i(x) q(x) dx = 0 \quad \text{for } i = 1, 2, \ldots, b. \tag{4}$$

The function $r(x)$ governs the nature of the model error, and δ is the possible magnitude of this error. In order to separate these two factors, we further impose the following normalization condition on $r(x)$:

$$\int r^2(x) q(x) dx = 1. \tag{5}$$

Note that we are essentially estimating the projection $g(x)$, rather than the true target function $f(x)$.

2. Note that we do not impose any restrictions on the choice of basis functions. Therefore, Eq.(2) includes a variety of models such as polynomial models, trigonometric polynomial models, and Gaussian kernel models with fixed centers.

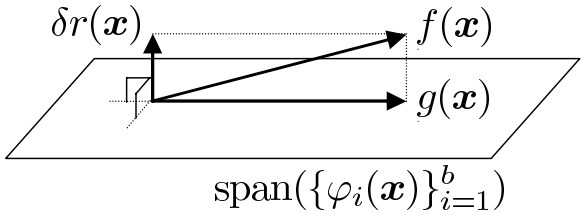

Figure 3: Orthogonal decomposition of $f(x)$.

2.3 Bias/Variance Decomposition of Generalization Error

As described in Section 1, we evaluate the generalization error in terms of the expectation over only the training output noise $\{\varepsilon_i\}_{i=1}^n$, not over the training input points $\{x_i\}_{i=1}^n$.

Let $\mathbb{E}_{\{\varepsilon_i\}}$ denote the expectation over the noise $\{\varepsilon_i\}_{i=1}^n$. Then, the generalization error expected over the training output noise can be decomposed into the (squared) *bias* term B, the *variance* term V, and the model error C:

$$\mathop{\mathbb{E}}_{\{\varepsilon_i\}} G' = B + V + C,$$

where

$$\begin{aligned} B &= \int \left(\mathop{\mathbb{E}}_{\{\varepsilon_i\}} \widehat{f}(x) - g(x) \right)^2 q(x) dx, \\ V &= \mathop{\mathbb{E}}_{\{\varepsilon_i\}} \int \left(\widehat{f}(x) - \mathop{\mathbb{E}}_{\{\varepsilon_i\}} \widehat{f}(x) \right)^2 q(x) dx, \\ C &= \int (g(x) - f(x))^2 q(x) dx. \end{aligned} \tag{6}$$

Since C is constant which depends neither on $p(x)$ nor $\{x_i\}_{i=1}^n$, we subtract C from G' and define it by G.

$$G = G' - C.$$

2.4 Importance-Weighted Least-Squares Learning

Let X be the *design matrix*, i.e., X is the $n \times b$ matrix with the (i, j)-th element

$$X_{i,j} = \varphi_j(x_i).$$

A standard way to learn the parameters in the regression model (2) is the *ordinary least-squares (OLS) learning*, i.e., parameter vector α is determined as follows.

$$\widehat{\alpha}_O = \mathop{\mathrm{argmin}}_{\alpha} \left[\sum_{i=1}^n \left(\widehat{f}(x_i) - y_i \right)^2 \right], \tag{7}$$

where the subscript 'O' indicates the ordinary LS. $\widehat{\alpha}_O$ is analytically given by

$$\widehat{\alpha}_O = L_O y,$$

where

$$L_O = (X^\top X)^{-1} X^\top,$$
$$y = (y_1, y_2, \ldots, y_n)^\top.$$

When the training input points $\{x_i\}_{i=1}^n$ are drawn from $q(x)$, OLS is asymptotically unbiased even for misspecified models. However, the current situation is under the *covariate shift* (Shimodaira, 2000), where the training input density $p(x)$ is generally different from the test input density $q(x)$. Under the covariate shift, OLS is no longer unbiased even asymptotically for misspecified models. On the other hand, it is known that the following *weighted least-squares (WLS) learning* is asymptotically unbiased (Shimodaira, 2000).

$$\widehat{\alpha}_W = \operatorname*{argmin}_{\alpha} \left[\sum_{i=1}^{n} \frac{q(x_i)}{p(x_i)} \left(\widehat{f}(x_i) - y_i \right)^2 \right], \tag{8}$$

where the subscript 'W' indicates the weighted LS. Asymptotic unbiasedness of $\widehat{\alpha}_W$ would be intuitively understood by the following identity, which resembles the *importance sampling*:

$$\int \left(\widehat{f}(x) - f(x) \right)^2 q(x) dx = \int \left(\widehat{f}(x) - f(x) \right)^2 \frac{q(x)}{p(x)} p(x) dx.$$

In the following, we assume that $p(x)$ and $q(x)$ are strictly positive for all x.

Let D be the diagonal matrix with the i-th diagonal element

$$D_{i,i} = \frac{q(x_i)}{p(x_i)}.$$

Then $\widehat{\alpha}_W$ is analytically given by

$$\widehat{\alpha}_W = L_W y, \tag{9}$$

where

$$L_W = (X^\top D X)^{-1} X^\top D.$$

2.5 Active Learning Based on Importance-Weighted Least-Squares Learning

Let G_W, B_W and V_W be G, B and V for the learned function obtained by WLS, respectively. Let U be the b-dimensional square matrix with the (i, j)-th element

$$U_{i,j} = \int \varphi_i(x) \varphi_j(x) q(x) dx.$$

Then we have the following lemma (Proofs of all lemmas are provided in appendices).

Lemma 1 *For the approximately correct model (3), we have*

$$B_W = O_p(\delta^2 n^{-1}),$$
$$V_W = \sigma^2 \mathrm{tr}(U L_W L_W^\top) = O_p(n^{-1}). \tag{10}$$

Input: A finite set $\widehat{\mathcal{P}}$ of strictly positive probability densities

Calculate U.
For each $p \in \widehat{\mathcal{P}}$
    Create training input points $\{x_i^{(p)}\}_{i=1}^n$ following $p(x)$.
    Calculate L_W.
    Calculate $J(p)$.
End
Choose $\widehat{p}$ that minimizes J.
Put $x_i = x_i^{(\widehat{p})}$ for $i = 1, 2, \ldots, n$.
Observe the training output values $\{y_i\}_{i=1}^n$ at $\{x_i\}_{i=1}^n$.
Calculate $\widehat{\alpha}_W$ by Eq.(9).

Output: $\widehat{\alpha}_W$

Figure 4: Proposed ALICE algorithm.

Note that the asymptotic order in the above lemma is in probability since random variables $\{x_i\}_{i=1}^n$ are included. This lemma implies that if $\delta = o_p(1)$,

$$\mathop{\mathbb{E}}_{\{\varepsilon_i\}} G_W = \sigma^2 \mathrm{tr}(UL_W L_W^\top) + o_p(n^{-1}). \tag{11}$$

Motivated by Eq.(11), we propose determining the training input density $p(x)$ as follows: For a set $\mathcal{P}$ of strictly positive probability densities,

$$p^* = \mathop{\mathrm{argmin}}_{p \in \mathcal{P}} J(p),$$

where

$$J = \mathrm{tr}(UL_W L_W^\top). \tag{12}$$

Practically, we may prepare a finite set $\widehat{\mathcal{P}}$ of strictly positive probability densities and choose the one that minimizes J from the set $\widehat{\mathcal{P}}$. A pseudo code of the proposed active learning algorithm is described in Figure 4, which we call *ALICE* (Active Learning using the Importance-weighted least-squares learning based on Conditional Expectation of the generalization error). Note that the value of J depends not only on $p(x)$, but also on the realization of the training input points $\{x_i^{(p)}\}_{i=1}^n$.

3. Relation to Existing Methods

In this section, we qualitatively compare the proposed active learning method with existing methods.

3.1 Active Learning with OLS

Let G_O, B_O and V_O be G, B and V for the learned function obtained by OLS, respectively. If $\delta = 0$ in Eq.(3), i.e., the model is correctly specified, B_O vanishes under a mild condition (Fedorov, 1972) and we have

$$\mathop{\mathbb{E}}_{\{\varepsilon_i\}} G_O = V_O = \sigma^2 \mathrm{tr}(UL_O L_O^\top).$$

Based on the above expression, the training input density $p(x)$ is determined[3] as follows (Fedorov, 1972; Cohn et al., 1996; Fukumizu, 2000).

$$p_O^* = \operatorname*{argmin}_{p \in \mathcal{P}} J_O(p),$$

where

$$J_O = \mathrm{tr}(U L_O L_O^\top). \tag{13}$$

Comparison with J: We investigate the validity of J_O for approximately correct models based on the following lemma.

Lemma 2 *For the approximately correct model (3), we have*

$$B_O = O(\delta^2),$$
$$V_O = O_p(n^{-1}).$$

The above lemma implies that if $\delta = o_p(n^{-\frac{1}{2}})$,

$$\mathop{\mathbb{E}}_{\{\varepsilon_i\}} G_O = \sigma^2 J_O + o_p(n^{-1}).$$

Therefore, if $\delta = o_p(n^{-\frac{1}{2}})$, the use of J_O can be still justified. On the other hand, the proposed J is valid when $\delta = o_p(1)$. This implies that J has a wider range of applications than J_O. As experimentally shown in Section 4, this difference is highly significant in practice.

3.2 Active Learning with WLS: Variance-Only Approach

For the importance-weighted least-squares learning (8), Kanamori and Shimodaira (2003) proved that the generalization error expected over training input points $\{x_i\}_{i=1}^n$ and training output noise $\{\varepsilon_i\}_{i=1}^n$ is asymptotically expressed as

$$\mathop{\mathbb{E}}_{\{x_i\}} \mathop{\mathbb{E}}_{\{\varepsilon_i\}} G_W = \frac{1}{n}\mathrm{tr}(U^{-1}H) + O(n^{-\frac{3}{2}}), \tag{14}$$

where $\mathbb{E}_{\{x_i\}}$ is the expectation over training input points $\{x_i\}_{i=1}^n$ and H is the b-dimensional square matrix defined by

$$H = S + \sigma^2 T.$$

S and T are the b-dimensional square matrices with the (i,j)-th elements

$$S_{i,j} = \int \varphi_i(x)\varphi_j(x)(\delta r(x))^2 \frac{q(x)^2}{p(x)} dx, \tag{15}$$

$$T_{i,j} = \int \varphi_i(x)\varphi_j(x) \frac{q(x)^2}{p(x)} dx. \tag{16}$$

3. $p(x)$ is not explicitly used in OLS. Therefore, we do not have to optimize the training input density $p(x)$, but we can directly optimize training input points $\{x_i\}_{i=1}^n$. However, to be consistent with the WLS-based methods, we optimize $p(x)$ in this paper. This also helps to avoid the simultaneous optimization of n input points which is computationally very demanding in general.

Note that $\frac{1}{n}\mathrm{tr}(U^{-1}S)$ corresponds to the squared bias while $\frac{\sigma^2}{n}\mathrm{tr}(U^{-1}T)$ corresponds to the variance. Eq.(14) suggests that $\mathrm{tr}(U^{-1}H)$ may be used as an active learning criterion. However, H includes the inaccessible quantities $\delta r(x)$ and σ^2, so $\mathrm{tr}(U^{-1}H)$ can not be directly calculated.

To cope with this problem, Wiens (2000) proposed[4] ignoring S (the bias term), which yields

$$\mathop{\mathbb{E}}_{\{x_i\}}\mathop{\mathbb{E}}_{\{\varepsilon_i\}} G_W \approx \frac{\sigma^2}{n}\mathrm{tr}(U^{-1}T).$$

Note that T is accessible under the current setting. Based on this approximation, the training input density $p(x)$ is determined as follows.

$$p_W^* = \mathop{\mathrm{argmin}}_{p\in\mathcal{P}} J_W(p),$$

where

$$J_W = \frac{1}{n}\mathrm{tr}(U^{-1}T). \tag{17}$$

Comparison with J: A notable feature of J_W is that the optimal training input density $p_W^*(x)$ can be obtained analytically (Wiens, 2000):

$$p_W^*(x) = \frac{\widehat{h}(x)}{\int \widehat{h}(x)dx}, \tag{18}$$

where

$$\widehat{h}(x) = q(x)\left(\sum_{i,j=1}^{b} U_{i,j}^{-1}\varphi_i(x)\varphi_j(x)\right)^{\frac{1}{2}}.$$

This may be confirmed by the fact that J_W can be expressed as

$$J_W(p) = \frac{1}{n}\left(\int \widehat{h}(x)dx\right)^2\left(1+\int\frac{(p_W^*(x)-p(x))^2}{p(x)}dx\right).$$

On the other hand, we do not yet have an analytic form of a minimizer for the criterion J.

It seems that in Wiens (2000), ignoring S has not been well justified. Here, we investigate the validity based on the following corollary immediately obtained from Eqs.(14) and (15).

Corollary 1 *For the approximately correct model (3), we have*

$$\mathop{\mathbb{E}}_{\{x_i\}}\mathop{\mathbb{E}}_{\{\varepsilon_i\}} G_W = \sigma^2 J_W + O(\delta^2 n^{-1} + n^{-\frac{3}{2}}),$$

where $\sigma^2 J_W = O(n^{-1})$.

4. In the original paper, discussion is restricted to the cases where the input domain is bounded and $q(x)$ is uniform over the domain. However, it may be easily extended to an arbitrary strictly-positive $q(x)$. For this reason, we deal with the extended version here.

This corollary implies that if $\delta = o(1)$,

$$\mathop{\mathbb{E}}_{\{x_i\}} \mathop{\mathbb{E}}_{\{\varepsilon_i\}} G_W = \sigma^2 J_W + o(n^{-1}),$$

by which the use of J_W can be justified asymptotically. Since the order is the same as that of the proposed criterion, J and J_W may be comparable in the robustness against the misspecification of models.

Now the following lemma reveals a more direct relation between J and J_W.

Lemma 3 *J and J_W satisfy*

$$J = J_W + O_p(n^{-\frac{3}{2}}). \tag{19}$$

This lemma implies that J is asymptotically equivalent to J_W. However, they are still different in the order of n^{-1}. In the following, we show that this difference is important.

In the active learning context, we are interested in accurately predicting the *single-trial* generalization error G_W, which depends on the realization of the training examples. Let us measure the goodness of a generalization error predictor $\widehat{G}$ by

$$\mathop{\mathbb{E}}_{\{\varepsilon_i\}} (\widehat{G} - G_W)^2. \tag{20}$$

Then we have the following lemma.

Lemma 4 *Suppose $\delta = o_p(n^{-\frac{1}{4}})$. If terms of $o_p(n^{-3})$ are ignored, we have*

$$\mathop{\mathbb{E}}_{\{\varepsilon_i\}} (\sigma^2 J_W - G_W)^2 \geq \mathop{\mathbb{E}}_{\{\varepsilon_i\}} (\sigma^2 J - G_W)^2.$$

This lemma states that under $\delta = o_p(n^{-\frac{1}{4}})$, $\sigma^2 J$ is asymptotically a more accurate estimator of the single-trial generalization error G_W than $\sigma^2 J_W$ in the sense of Eq.(20).

In Section 4, we experimentally evaluate the difference between J and J_W.

3.3 Active Learning with WLS: Bias-and-Variance Approach

Another idea of approximating H in Eq.(14) is a two-stage sampling scheme proposed[5] by Kanamori and Shimodaira (2003): the training examples sampled in the first stage are used for estimating H and in the second stage, the distribution of the remaining training input points is optimized based on the estimated H. We explain the details of the algorithm below.

First, ℓ ($\leq n$) training input points $\{\widetilde{x}_i\}_{i=1}^{\ell}$ are created independently following the test input distribution with density $q(x)$, and corresponding training output values $\{\widetilde{y}_i\}_{i=1}^{\ell}$ are observed. Let $\widetilde{D}$ and $\widetilde{Q}$ be the ℓ-dimensional diagonal matrices with the i-th diagonal elements

$$\widetilde{D}_{i,i} = \frac{q(\widetilde{x}_i)}{p(\widetilde{x}_i)},$$
$$\widetilde{Q}_{i,i} = [\widetilde{y} - \widetilde{X}(\widetilde{X}^\top \widetilde{X})^{-1} \widetilde{X}^\top \widetilde{y}]_i,$$

5. In the original paper, the method is derived within a slightly different setting of estimating the conditional probability of the output value y given an input point x for regular statistical models. Here, we focus on the cases where the conditional distirbution is Gaussian and the statistical model is linear, by which the setting becomes comparable to that of the current paper.

where $[\cdot]_i$ denotes the i-th element of a vector. $\widetilde{X}$ is the design matrix for $\{\widetilde{x}_i\}_{i=1}^{\ell}$, i.e., the $\ell \times b$ matrix with the (i,j)-th element

$$\widetilde{X}_{i,j} = \varphi_j(\widetilde{x}_i),$$

and

$$\widetilde{y} = (\widetilde{y}_1, \widetilde{y}_2, \ldots, \widetilde{y}_\ell)^\top.$$

Then an approximation $\widetilde{H}$ of the unknown matrix H in Eq.(14) is given by

$$\widetilde{H} = \frac{1}{\ell}\widetilde{X}^\top \widetilde{D}\widetilde{Q}^2\widetilde{X}.$$

Although U^{-1} is accessible in the current setting, Kanamori and Shimodaira (2003) also replaced it by a consistent estimate $\widetilde{U}^{-1}$, where

$$\widetilde{U} = \frac{1}{\ell}\widetilde{X}^\top \widetilde{X}.$$

Based on the above approximations, the training input density $p(x)$ is determined as follows:

$$p^*_{OW} = \operatorname*{argmin}_{p \in \mathcal{P}} J_{OW}(p),$$

where

$$J_{OW} = \frac{1}{n}\mathrm{tr}(\widetilde{U}^{-1}\widetilde{H}). \tag{21}$$

Note that the subscript '*OW*' indicates the combination of the ordinary LS and weighted LS (see below for details).

After determining the optimal density p^*_{OW}, the remaining $n-\ell$ training input points $\{x_i\}_{i=1}^{n-\ell}$ are created independently following $p^*_{OW}(x)$, and corresponding training output values $\{y_i\}_{i=1}^{n-\ell}$ are observed. Then the learned parameter $\widehat{\alpha}_{OW}$ is obtained using $\{(\widetilde{x}_i, \widetilde{y}_i)\}_{i=1}^{\ell}$ and $\{(x_i, y_i)\}_{i=1}^{n-\ell}$ as

$$\widehat{\alpha}_{OW} = \operatorname*{argmin}_{\alpha} \left[\sum_{i=1}^{\ell} \left(\widehat{f}(\widetilde{x}_i) - \widetilde{y}_i \right)^2 + \sum_{i=1}^{n-\ell} \frac{q(x_i)}{p(x_i)} \left(\widehat{f}(x_i) - y_i \right)^2 \right]. \tag{22}$$

Note that J_{OW} depends on the realization of $\{\widetilde{x}_i\}_{i=1}^{\ell}$, but is independent of the realization of $\{x_i\}_{i=1}^{n-\ell}$.

Comparison with J: Kanamori and Shimodaira (2003) proved that for $\ell = o(n)$, $\lim_{n\to\infty} \ell = \infty$, and $\delta = O(1)$,

$$\mathop{\mathbb{E}}_{\{x_i\}} \mathop{\mathbb{E}}_{\{\varepsilon_i\}} G_W = \frac{1}{n} J_{OW} + o(n^{-1}),$$

by which the use of J_{OW} can be justified. The order of δ required above is weaker than that required in J. Therefore, J_{OW} may have a wider range of applications than J. However, this property may not be practically valuable since learning with totally misspecified models (i.e., $\delta = O(1)$) may not work well because of the model error.

Due to the two-stage sampling scheme, the above method has several weaknesses. First, ℓ training input points should be gathered following $q(x)$ in the first stage, which implies that users are only allowed to optimize the location of $n-\ell$ remaining training input points. This may be critical when the total number n is not so large. Second, the performance depends on the choice of ℓ, so it has to be appropriately determined. Using $\ell = O(n^{1/2})$ is recommended in Kanamori and

Shimodaira (2003), but the exact choice of ℓ seems still open. Third, J_{OW} is an estimator of G_W, but the finally obtained parameter by this algorithm is not $\widehat{\alpha}_W$ but $\widehat{\alpha}_{OW}$. Therefore, this difference can degrade the performance.[6]

In Section 4, we experimentally compare J and J_{OW}.

4. Numerical Examples

In this section, we quantitatively compare the proposed and existing active learning methods through numerical experiments.

4.1 Toy Data Set

We first illustrate how the proposed and existing methods behave under a controlled setting.

Setting: Let the input dimension be $d = 1$ and the learning target function be

$$f(x) = 1 - x + x^2 + \delta r(x),$$

where

$$r(x) = \frac{z^3 - 3z}{\sqrt{6}} \quad \text{with} \quad z = \frac{x - 0.2}{0.4}. \tag{23}$$

Let the number of training examples to gather be $n = 100$ and $\{\varepsilon_i\}_{i=1}^n$ be i.i.d. Gaussian noise with mean zero and standard deviation 0.3. Let the test input density $q(x)$ be the Gaussian density with mean 0.2 and standard deviation 0.4, which is assumed to be known in this illustrative simulation. See the bottom graph of Figure 5 for the profile of $q(x)$. Let the number of basis functions be $b = 3$ and the basis functions be

$$\varphi_i(x) = x^{i-1} \quad \text{for } i = 1, 2, \ldots, b.$$

Note that for these basis functions, the residual function $r(x)$ in Eq.(23) fulfills Eqs.(4) and (5). Let us consider the following three cases.

$$\delta = 0, 0.005, 0.05, \tag{24}$$

which correspond to "*correctly specified*", "*approximately correct*", and "*misspecified*" cases, respectively. See the top graph of Figure 5 for the profiles of $f(x)$ with different δ.

As a set of training input densities, $\widehat{\mathcal{P}}$, we use the Gaussian densities with mean 0.2 and standard deviation $0.4c$, where

$$c = 0.8, 0.9, 1.0, \ldots, 2.5.$$

See the bottom graph of Figure 5 again for the profiles of $p(x)$ with different c.

In this experiment, we compare the performance of the following methods:

(ALICE): c is determined so that J given by Eq.(12) is minimized. WLS given by Eq.(8) is used for estimating the parameters.

6. It is possible to resolve this problem by not using $\{(\widetilde{x}_i, \widetilde{y}_i)\}_{i=1}^{\ell}$ gathered in the first stage for estimating the parameter (cf. Eq.(22)). However, this may yield further degradation of the performance because only $n - \ell$ training examples are used for learning.

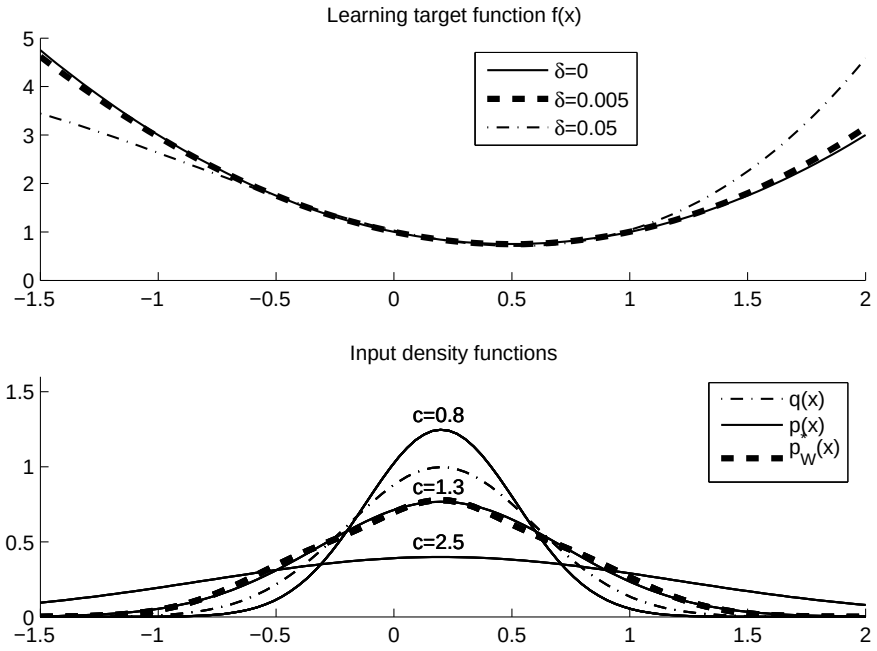

Figure 5: Learning target function and input density functions.

(W): c is determined so that J_W given by Eq.(17) is minimized. WLS is used for estimating the parameters.

(W*): $p^*_W(x)$ given by Eq.(18) is used as the training input density. The profile of $p^*_W(x)$ under the current setting is illustrated in the bottom graph of Figure 5, showing that $p^*_W(x)$ is similar to the Gaussian density with $c = 1.3$. WLS is used for estimating the parameters.

(OW): First, ℓ training input points are created following the test input density $q(x)$, and corresponding training output values are observed. Based on the ℓ training examples, c is determined so that J_{OW} given by Eq.(21) is minimized. Then $n-\ell$ remaining training input points are created following the determined input density. The combination of OLS and WLS given by Eq.(22) is used for estimating the parameters. We set $\ell = 25$, which we experimentally confirmed to be a reasonable choice in this illustrative simulation.

(O): c is determined so that J_O given by Eq.(13) is minimized. OLS given by Eq.(7) is used for estimating the parameters.

(Passive): Following the test input density $q(x)$, training input points $\{x_i\}_{i=1}^n$ are created. OLS is used for estimating the parameters.

For (W*), we generate the random number following $p^*_W(x)$ by the rejection method (see e.g., Knuth, 1998). We run this simulation 1000 times for each δ in Eq.(24).

Accuracy of Generalization Error Prediction: First, we evaluate the accuracy of J, J_W, J_{OW}, and J_O as predictors of the generalization error. Note that J and J_W are predictors of G_W. J_{OW} is also derived as a predictor of G_W, but the finally obtained generalization error by (OW) is G_{OW}, which

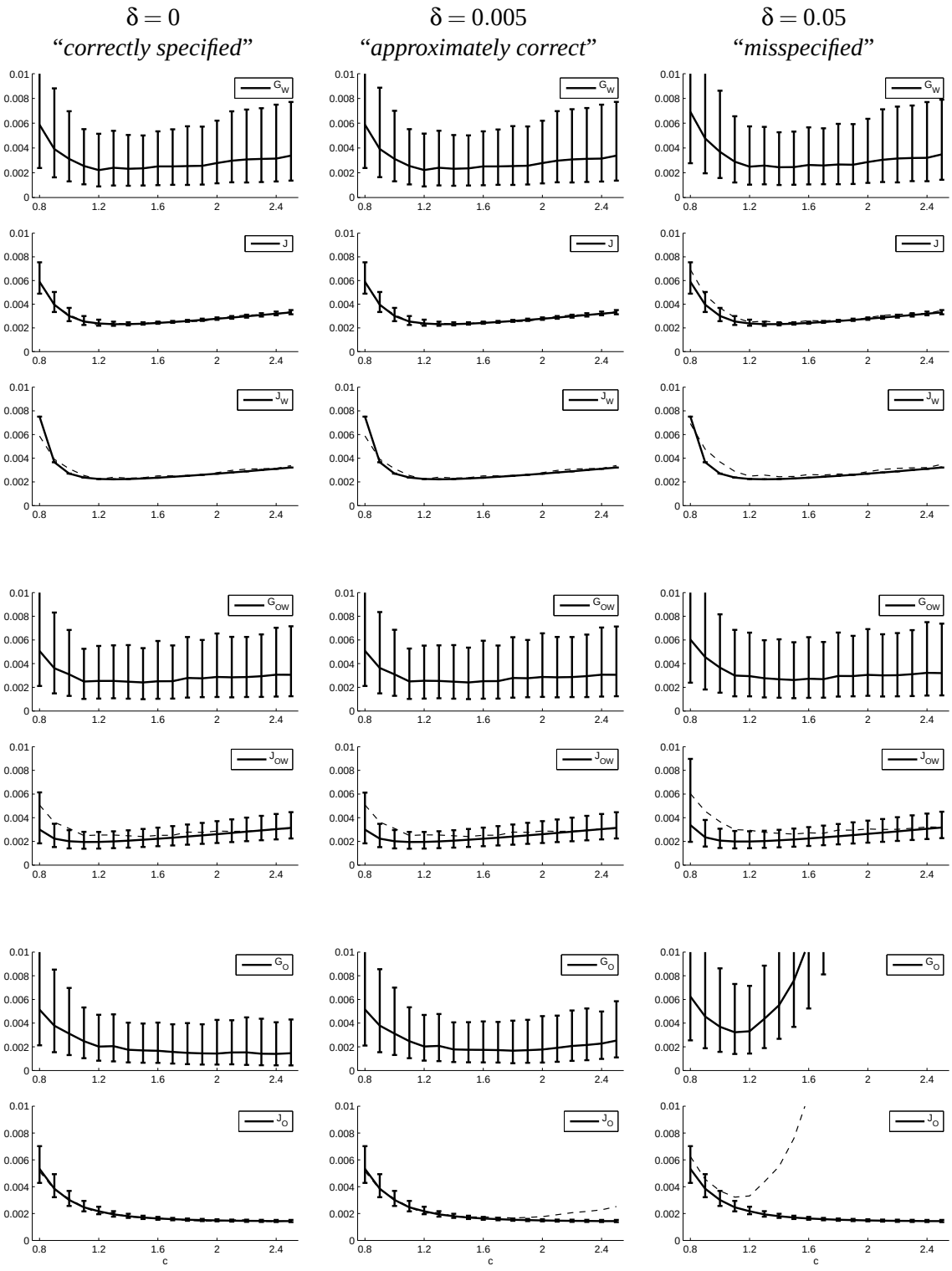

Figure 6: The means and (asymmetric) standard deviations of G_W, J, J_W, G_{OW}, J_{OW}, G_O, and J_O over 1000 runs as functions of c. The dashed curves show the means of the generalization error that corresponding active learning criteria are trying to predict.

is the generalization error G for the learned function obtained by the combination of OLS and WLS (see Eq.(22)). Therefore, J_{OW} should be evaluated as a predictor of G_{OW}. J_O is a predictor of G_O.

In Figure 6, the means and standard deviations of G_W, J, J_W, G_{OW}, J_{OW}, G_O, and J_O over 1000 runs are depicted as functions of c by the solid curves. Here the upper and lower error bars are calculated separately since the distribution is not symmetric. The dashed curves show the means of the generalization error that corresponding active learning criteria are trying to predict. Note that J, J_W, and J_O are multiplied by $\sigma^2 = (0.3)^2$ so that comparison with G_W and G_O are clear. By definition, G_W, G_{OW}, and G_O do not include the constant C defined by Eq.(6). The values of C for $\delta = 0$, 0.005, and 0.05 are 0, 2.32×10^{-5}, and 2.32×10^{-3}, respectively.

These graphs show that when $\delta = 0$ ("*correctly specified*"), J and J_W give accurate predictions of G_W. Note that J_W does not depend on the training input points $\{x_i\}_{i=1}^n$ so it does not fluctuate over 1000 runs. J_{OW} is slightly biased toward the negative direction for small c. We conjecture that this is caused by the small sample effect. However, the profile of J_{OW} still roughly approximates that of G_{OW}. J_O gives accurate predictions of G_O. When $\delta = 0.005$ ("*approximately correct*"), J, J_W, and J_{OW} work similarly to the case with $\delta = 0$, i.e., J and J_W are accurate and J_{OW} is negatively biased. On the other hand, J_O behaves slightly differently: it tends to be biased toward the negative direction for large c. Finally, when $\delta = 0.05$ ("*misspecified*"), J and J_W still give accurate predictions, although they slightly have a negative bias for small c. J_{OW} still roughly approximates G_{OW}, while J_O gives totally different profile from G_O.

These results show that as approximations of the generalization error, J and J_W are accurate and robust against the misspecification of models. J_{OW} is also reasonably accurate, although it tends to be rather inaccurate for small c. J_O is accurate in the correctly specified case, but it becomes totally inaccurate once the correctness of the model is violated.

Note that, by definition, J, J_W and J_O do not depend on the learning target function. Therefore, in the simulation, they give the same values for all δ (J and J_O depend on the realization of $\{x_i\}_{i=1}^n$ so they may have a small fluctuation). On the other hand, the generalization error, of course, depends on the learning target function even if the constant C is not included, since the training output values depend on it. Note that the bias depends on δ, but the variance does not. The simulation results show that the profile of G_O changes heavily as the degree of model misspecification increases. This would be caused by the increase of the bias since OLS is not unbiased even asymptotically. On the other hand, J_O stays the same as δ increases. As a result, J_O becomes a very poor predictor for a large δ. In contrast, the profile of G_W appears to be very stable against the change in δ, which is in good agreement with the theoretical fact that WLS is asymptotically unbiased. Thanks to this property, J and J_W are more accurate than J_O for misspecified models.

Obtained Generalization Error: In Table 1, the mean and standard deviation of the generalization error obtained by each method are described. The best method and comparable ones by the *t-test* (e.g., Henkel, 1979) at the significance level 5% are indicated with boldface. In Figure 7, the box-plot expression of the obtained generalization error is depicted. Note that the values described in Figure 6 correspond to G (the constant C is not included), while the values in Table 1 and Figure 7 correspond to G' which includes C (see Eq.(1)).

When $\delta = 0$, (O) works significantly better than other methods. Actually, in this case, training input densities that approximately minimize G_W, G_O, and G_{OW} were successfully found by (ALICE), (W), (OW), and (O). This implies that the difference in the error is caused not by the quality of the active learning criteria, but by the difference between WLS and OLS: WLS generally has

	$\delta = 0$	$\delta = 0.005$	$\delta = 0.05$
(ALICE)	2.08 ± 1.95	$\mathbf{2.10 \pm 1.96}$	$\mathbf{4.61 \pm 2.12}$
(W)	2.40 ± 2.15	2.43 ± 2.15	4.89 ± 2.26
(W*)	2.32 ± 2.02	2.35 ± 2.02	4.84 ± 2.14
(OW)	3.09 ± 3.03	3.13 ± 3.00	5.95 ± 3.58
(O)	$\mathbf{1.31 \pm 1.70}$	2.53 ± 2.23	124 ± 67.4
(Passive)	3.11 ± 2.78	3.14 ± 2.78	6.01 ± 3.43

All values in the table are multiplied by 10^3.

Table 1: The mean and standard deviation of the generalization error obtained by each method for the toy data set. Here we describe the value G' that includes the constant C (see Eq.(6)). The best method and comparable ones by the t-test at the significance level 5% are indicated with boldface. The value of (O) for $\delta = 0.05$ is extremely large but it is not a typo.

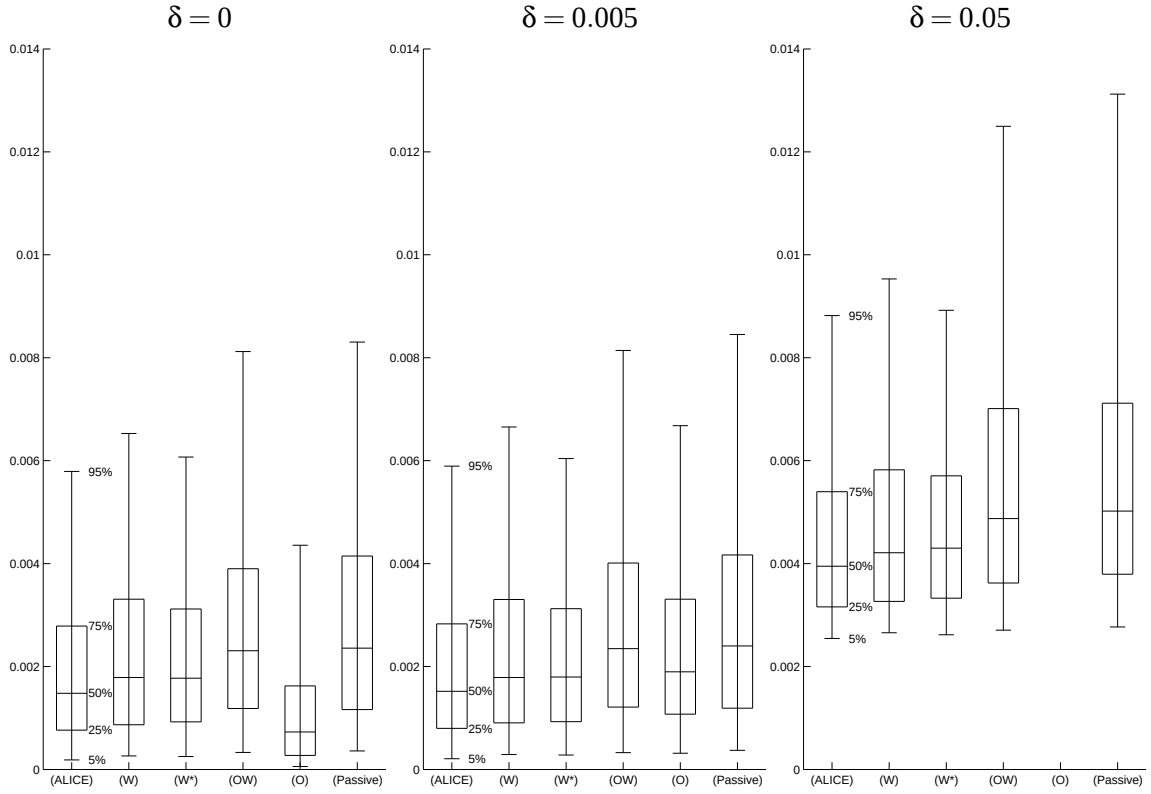

Figure 7: Box-plots of the generalization error obtained by each method for the toy data set. Here we plot the value G' that includes the constant C (see Eq.(6)). The value of (O) for $\delta = 0.05$ is not plotted because it is extremely large.

larger variance than OLS (Shimodaira, 2000). Therefore, when $\delta = 0$, OLS would be more accurate than WLS since both WLS and OLS are unbiased. Although (ALICE), (W), (W*), and (OW) are outperformed by (O), they still work better than (Passive). Note that (ALICE) is significantly better than (W), (W*), (OW), and (Passive) by the t-test. The box-plot shows that (ALICE) outperforms (W), (W*), and (OW) particularly in upper quantiles.

When $\delta = 0.005$, (ALICE) gives significantly smaller errors than other methods. All the methods except (O) work similarly to the case with $\delta = 0$, while (O) tends to perform poorly. This result is surprising since the learning target functions with $\delta = 0$ and $\delta = 0.005$ are visually almost the same, as illustrated in the top graph of Figure 5. Therefore, it intuitively seems that the result when $\delta = 0.005$ is not much different from the result when $\delta = 0$. However, this slight difference appears to make (O) unreliable.

When $\delta = 0.05$, (ALICE) again works significantly better than others. (W) and (W*) still work reasonably well. The box-plot shows that (ALICE) is better than (W) and (W*) particularly in upper quantiles. The performance of (OW) is slightly degraded, although it is still better than (Passive). (O) gives extremely large errors.

The above results are summarized as follows. For all three cases ($\delta = 0, 0.005, 0.05$), (ALICE), (W), (W*), and (OW) work reasonably well and consistently outperform (Passive). Among them, (ALICE) appears to be better than (W), (W*), and (OW) for all three cases. (O) works excellently in the correctly specified case, although it tends to perform poorly once the correctness of the model is violated. Therefore, (ALICE) is found to work well overall and is robust against the misspecification of models for this toy data set.

4.2 Benchmark Data Sets

Here we use eight regression benchmark data sets provided by DELVE (Rasmussen et al., 1996): *Bank-8fm, Bank-8fh, Bank-8nm, Bank-8nh, Kin-8fm, Kin-8fh, Kin-8nm,* and *Kin-8nh*. Each data set includes 8192 samples, consisting of 8-dimensional input points and 1-dimensional output values. For convenience, every attribute is normalized into $[0,1]$.

Suppose we are given all 8192 *input* points (i.e., unlabeled samples). Note that output values are kept unknown at this point. From this pool of unlabeled samples, we choose $n = 300$ input points $\{x_i\}_{i=1}^n$ for training and observe the corresponding output values $\{y_i\}_{i=1}^n$. The task is to predict the output values of all 8192 unlabeled samples.

In this experiment, the test input density $q(x)$ is unknown. So we estimate it using the uncorrelated multi-dimensional Gaussian density:

$$q(x) = \frac{1}{(2\pi\widehat{\gamma}_{MLE}^2)^{\frac{d}{2}}} \exp\left(-\frac{\|x-\widehat{\mu}_{MLE}\|^2}{2\widehat{\gamma}_{MLE}^2}\right),$$

where $\widehat{\mu}_{MLE}$ and $\widehat{\gamma}_{MLE}$ are the maximum likelihood estimates of the mean and standard deviation obtained from all 8192 unlabeled samples. Let $b = 50$ and the basis functions be Gaussian basis functions with variance 1:

$$\varphi_i(x) = \exp\left(-\frac{\|x-t_i\|^2}{2}\right) \qquad \text{for } i = 1,2,\ldots,b,$$

where $\{t_i\}_{i=1}^b$ are template points randomly chosen from the pool of unlabeled samples.

	Bank-8fm	Bank-8fh	Bank-8nm	Bank-8nh
(ALICE)	2.10 ± 0.17	6.83 ± 0.44	$\mathbf{1.11 \pm 0.09}$	4.19 ± 0.29
(W)	2.26 ± 0.21	7.21 ± 0.52	1.22 ± 0.12	4.40 ± 0.38
(OW)	2.31 ± 0.25	7.39 ± 0.63	1.25 ± 0.15	4.52 ± 0.39
(O)	$\mathbf{1.91 \pm 0.16}$	$\mathbf{6.20 \pm 0.24}$	1.32 ± 0.14	$\mathbf{4.02 \pm 0.21}$
(Passive)	2.31 ± 0.26	7.45 ± 0.61	1.26 ± 0.14	4.51 ± 0.38

	Kin-8fm	Kin-8fh	Kin-8nm	Kin-8nh
(ALICE)	$\mathbf{1.62 \pm 0.58}$	$\mathbf{3.50 \pm 0.63}$	$\mathbf{34.97 \pm 1.90}$	$\mathbf{47.21 \pm 1.97}$
(W)	$\mathbf{1.70 \pm 0.62}$	$\mathbf{3.64 \pm 0.73}$	36.60 ± 2.05	49.15 ± 2.88
(OW)	$\mathbf{1.73 \pm 0.63}$	3.73 ± 0.78	37.29 ± 2.94	49.64 ± 3.11
(O)	3.03 ± 1.60	4.85 ± 1.96	38.65 ± 3.09	48.86 ± 2.66
(Passive)	$\mathbf{1.77 \pm 0.68}$	3.73 ± 0.79	37.38 ± 3.05	49.69 ± 3.06

All values in the table are multiplied by 10^3.

Table 2: The means and standard deviations of the test error for DELVE data sets. The best method and comparable ones by the t-test at the significance level 5% are indicated with boldface.

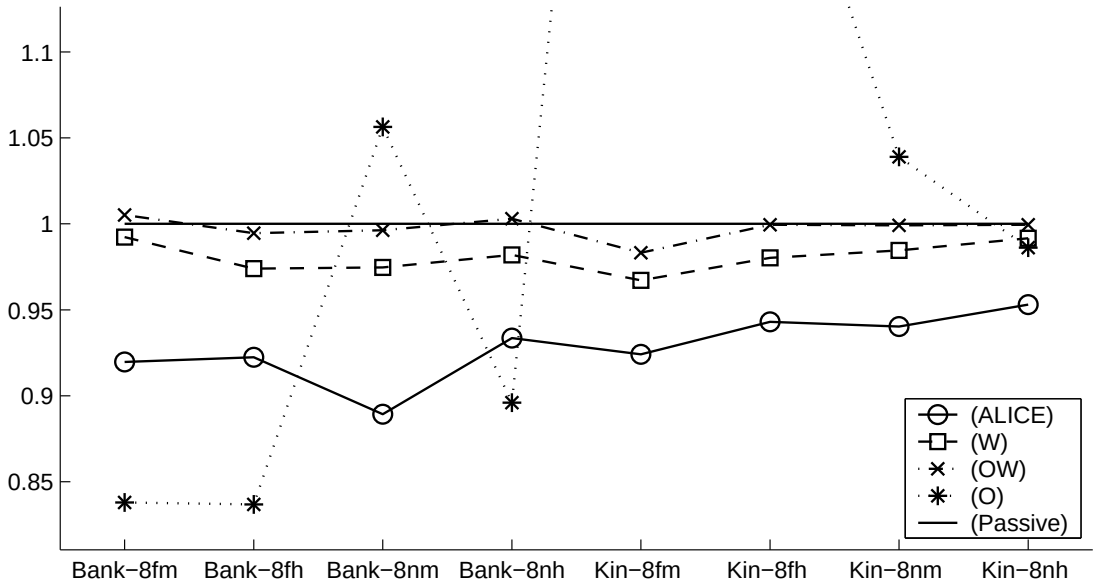

Figure 8: The means of the test error of (ALICE), (W), (OW), and (O) normalized by the test error of (Passive).

We select the training input density $p(x)$ from the set of uncorrelated multi-dimensional Gaussian densities with mean $\widehat{\mu}_{MLE}$ and standard deviation $c\widehat{\gamma}_{MLE}$, where

$$c = 0.7, 0.75, 0.8, \ldots, 2.4.$$

We again compare the active learning methods tested in Section 4.1. However, we do not test (W*) here because we could not efficiently generate random numbers following $p^*_W(x)$ by the rejection method. For (OW), we set $\ell = 100$ which we experimentally confirmed to be reasonable.

In this simulation, we can not create the training input points in an arbitrary location because we only have 8192 samples in the pool. Here, we first create provisional input points following the determined training input density, and then choose the input points from the pool of unlabeled samples that are closest to the provisional input points. In this simulation, the expectation over the test input density $q(x)$ in the matrix U is calculated by the empirical average over all 8192 unlabeled samples since the true test error is also calculated as such. For each data set, we run this simulation 100 times, by changing the template points $\{t_i\}_{i=1}^b$ in each run.

The means and standard deviations of the test error over 100 runs are described in Table 2. This shows that (ALICE) works very well for five out of eight data sets. For the other three data sets, (O) works significantly better than other methods. (W) works well and is comparable to (ALICE) for two data sets, but is outperformed by (ALICE) for the other six data sets. (OW) is overall comparable to (Passive).

Figure 8 depicts the means of the test error of (ALICE), (W), (OW), and (O) normalized by the test error of (Passive): For each run, the test errors of (ALICE), (W), (OW), and (O) are divided by the test error of (Passive), and then the values are averaged over 100 runs. This graph shows that (ALICE) is better than (W), (OW), and (Passive) for all eight data sets. (O) works very well for three data sets, but it is comparable or largely outperformed by (Passive) for the other five data sets. (W) also works reasonably well, although it is outperformed by (ALICE) overall. (OW) is on par with (Passive). Overall, (ALICE) is shown to be stable and works well for the benchmark data sets.

We also carried out similar simulations for Gaussian basis functions with variance 0.5 and 2. The results had similar tendencies, i.e., (ALICE) is overall shown to be stable and works well, so we omit the detail.

5. Conclusions

In this paper, we proposed a new active learning method based on the importance-weighted least-squares learning. The numerical study showed that the proposed method works well overall and compares favorably to existing WLS-based methods and the passive learning scheme. Although the proposed method is outperformed by the existing OLS-based method when the model is correctly specified, the existing OLS-based method tends to perform very poorly once the correctness of the model is violated. Therefore, the existing OLS-based method may not be reliable in practical situations where the correctness of the model may not be fulfilled. On the other hand, the proposed method is shown to be robust against the misspecification of models and therefore reliable.

Our criterion is shown to be a variant of the criterion proposed by Wiens (2000). Indeed, we showed that they are asymptotically equivalent. However, an important difference is that we predict the conditional expectation of the generalization error given training input points, while in Wiens (2000), the full expectation of the generalization error is predicted. As described in Section 1, the conditional-expectation approach conceptually gives a finer choice of the training input density

than the full-expectation approach. Theoretically, we proved that the proposed criterion is a better estimate of the single-trial generalization error than Wiens's criterion (see Section 3.2).

An advantage of Wiens's criterion is that the optimal training input density can be obtained analytically, while we do not yet have such an analytic solution for the proposed criterion. In the current paper, we resorted to a naive optimization scheme: prepare a finite set of input densities and choose the best one from the set. The performance of this naive optimization scheme depends heavily on the choice of the set of densities. In practice, using a set of input densities which consist of the optimal density analytically found by Wiens's criterion and its variants would be a reasonable choice. It is also important to devise a better optimization strategy for the proposed active learning criterion, which currently remains open.

In theory, we assumed that the test input density is known. However, this may not be satisfied in practice. In experiments with benchmark data sets, the test input density is indeed unknown and is approximated by a Gaussian density. Although the simulation results showed that the proposed method consistently outperforms the passive learning scheme (given unlabeled samples), a more detailed analysis should be carried out to see how approximating the test input density affects the performance.

We discussed the active learning problem for *weakly* misspecified models. A natural extension of the proposed method is to be applicable to *strongly* misspecified models, as achieved in Kanamori and Shimodaira (2003). However, when the model is totally misspecified, even learning with the optimal training input points may not work well because of the model error. In such cases, it is important to carry out *model selection* (Akaike, 1974; Schwarz, 1978; Rissanen, 1978; Vapnik, 1998). In most of the active learning research—including the current paper, the location of the training input points are designed for a *single* model at hand. That is, the model should have been chosen *before* active learning is carried out. However, in practice, we may want to select the model as well as the location of the training input points. Devising a method for simultaneously optimizing the model and the location of the training input points would therefore be a more important and promising future direction. In Sugiyama and Ogawa (2003), a method of *active learning with model selection* has been proposed for the trigonometric polynomial models. However, its range of application is rather limited. We expect that the results given in this paper form a solid basis for further pursuing this challenging issue.

Acknowledgments

The author would like to thank anonymous reviewers for their helpful comments, which highly helped him to improve the manuscript. Particularly, the normalization of the residual function is pointed out by one of the reviewers. He also acknowledges Dr. Motoaki Kawanabe for fruitful discussions on the accuracy of generalization error estimators. Special thanks also go to the members of Fraunhofer FIRST.IDA for their comments on various aspects of the proposed method when the author gave a talk at the seminar. This work is supported by MEXT (Grant-in-Aid for Young Scientists 17700142).

Appendix A. Proof of Lemma 1

A simple calculation yields that B and V are expressed as

$$B = \langle U(\mathop{\mathbb{E}}_{\{\varepsilon_i\}} \widehat{\alpha} - \alpha^*), \mathop{\mathbb{E}}_{\{\varepsilon_i\}} \widehat{\alpha} - \alpha^* \rangle,$$
$$V = \mathop{\mathbb{E}}_{\{\varepsilon_i\}} \langle U(\widehat{\alpha} - \mathop{\mathbb{E}}_{\{\varepsilon_i\}} \widehat{\alpha}), \widehat{\alpha} - \mathop{\mathbb{E}}_{\{\varepsilon_i\}} \widehat{\alpha} \rangle.$$

Let

$$z_g = (g(x_1), g(x_2), \ldots g(x_n))^\top,$$
$$z_r = (r(x_1), r(x_2), \ldots r(x_n))^\top.$$

By definition, it holds that

$$z_g = X\alpha^*.$$

Then we have

$$\begin{aligned}
\mathop{\mathbb{E}}_{\{\varepsilon_i\}} \widehat{\alpha}_W - \alpha^* &= L_W(z_g + \delta z_r) - \alpha^* \\
&= (\tfrac{1}{n} X^\top D X)^{-1} \tfrac{1}{n} X^\top D (X\alpha^* + \delta z_r) - \alpha^* \\
&= \delta (\tfrac{1}{n} X^\top D X)^{-1} \tfrac{1}{n} X^\top D z_r.
\end{aligned}$$

By the law of large numbers (Rao, 1965), we have

$$\begin{aligned}
\lim_{n\to\infty} [\tfrac{1}{n} X^\top D X]_{i,j} &= \lim_{n\to\infty} \left(\frac{1}{n} \sum_{k=1}^{n} \frac{q(x_k)}{p(x_k)} \varphi_i(x_k) \varphi_j(x_k) \right) \\
&= \int_{\mathcal{D}} \frac{q(x)}{p(x)} \varphi_i(x) \varphi_j(x) p(x) dx \\
&= O_p(1).
\end{aligned}$$

Furthermore, by the central limit theorem (Rao, 1965), it holds for sufficiently large n,

$$\begin{aligned}
[\tfrac{1}{n} X^\top D z_r]_i &= \frac{1}{n} \sum_{k=1}^{n} r(x_k) \varphi_i(x_k) \frac{q(x_k)}{p(x_k)} \\
&= \int_{\mathcal{D}} r(x) \varphi_i(x) \frac{q(x)}{p(x)} p(x) dx + O_p(n^{-\frac{1}{2}}) \\
&= O_p(n^{-\frac{1}{2}}),
\end{aligned}$$

where the last equality follows from Eq.(4). Therefore, we have

$$\begin{aligned}
B_W &= \langle U(\mathop{\mathbb{E}}_{\{\varepsilon_i\}} \widehat{\alpha}_W - \alpha^*), \mathop{\mathbb{E}}_{\{\varepsilon_i\}} \widehat{\alpha}_W - \alpha^* \rangle \\
&= O_p(\delta^2 n^{-1}).
\end{aligned}$$

It holds that $U = O_p(1)$ and

$$\begin{aligned}
L_W L_W^\top &= (\tfrac{1}{n} X^\top D X)^{-1} \tfrac{1}{n^2} X^\top D^2 X (\tfrac{1}{n} X^\top D X)^{-1} \\
&= O_p(n^{-1}).
\end{aligned}$$

Then we have

$$\begin{aligned} V_W &= \mathop{\mathbb{E}}_{\{\varepsilon_i\}} \langle U(\widehat{\alpha}_W - \mathop{\mathbb{E}}_{\{\varepsilon_i\}} \widehat{\alpha}_W), \widehat{\alpha}_W - \mathop{\mathbb{E}}_{\{\varepsilon_i\}} \widehat{\alpha}_W \rangle \\ &= \sigma^2 \mathrm{tr}(U L_W L_W^\top) \\ &= O_p(n^{-1}), \end{aligned}$$

which concludes the proof. ■

Appendix B. Proof of Lemma 2

It holds that

$$\begin{aligned} \mathop{\mathbb{E}}_{\{\varepsilon_i\}} \widehat{\alpha}_O - \alpha^* &= L_O(z_g + \delta z_r) - \alpha^* \\ &= (\tfrac{1}{n} X^\top X)^{-1} \tfrac{1}{n} X^\top (X\alpha^* + \delta z_r) - \alpha^* \\ &= \delta (\tfrac{1}{n} X^\top X)^{-1} \tfrac{1}{n} X^\top z_r. \end{aligned}$$

By the law of large numbers, we have

$$\begin{aligned} \lim_{n\to\infty} [\tfrac{1}{n} X^\top X]_{i,j} &= \lim_{n\to\infty} \left(\frac{1}{n} \sum_{k=1}^{n} \varphi_i(x_k) \varphi_j(x_k) \right) \\ &= \int_{\mathcal{D}} \varphi_i(x) \varphi_j(x) p(x) dx \\ &= O_p(1). \end{aligned}$$

Furthermore, by the central limit theorem, it holds for sufficiently large n,

$$\begin{aligned} [\tfrac{1}{n} X^\top z_r]_i &= \frac{1}{n} \sum_{k=1}^{n} r(x_k) \varphi_i(x_k) \\ &= \int_{\mathcal{D}} r(x) \varphi_i(x) p(x) dx + O_p(n^{-\frac{1}{2}}) \\ &= O_p(1). \end{aligned}$$

Therefore, we have

$$\begin{aligned} B_O &= \langle U(\mathop{\mathbb{E}}_{\{\varepsilon_i\}} \widehat{\alpha}_O - \alpha^*), \mathop{\mathbb{E}}_{\{\varepsilon_i\}} \widehat{\alpha}_O - \alpha^* \rangle \\ &= O_p(\delta^2). \end{aligned}$$

It holds that $U = O_p(1)$ and

$$\begin{aligned} L_O L_O^\top &= (\tfrac{1}{n} X^\top X)^{-1} \tfrac{1}{n^2} X^\top X (\tfrac{1}{n} X^\top X)^{-1} \\ &= O_p(n^{-1}). \end{aligned}$$

Then we have

$$\begin{aligned} V_O &= \mathop{\mathbb{E}}_{\{\varepsilon_i\}} \langle U(\widehat{\alpha}_O - \mathop{\mathbb{E}}_{\{\varepsilon_i\}} \widehat{\alpha}_O), \widehat{\alpha}_O - \mathop{\mathbb{E}}_{\{\varepsilon_i\}} \widehat{\alpha}_O \rangle \\ &= \sigma^2 \mathrm{tr}(U L_O L_O^\top) \\ &= O_p(n^{-1}), \end{aligned}$$

which concludes the proof. ■

Appendix C. Proof of Lemma 3

The central limit theorem (see e.g., Rao, 1965) asserts that

$$L_W L_W^\top = \tfrac{1}{n} U^{-1} T U^{-1} + O_p(n^{-\frac{3}{2}}),$$

from which we have Eq.(19) ■

Appendix D. Proof of Lemma 4

It holds that

$$\begin{aligned}
\mathop{\mathbb{E}}_{\{\varepsilon_i\}} (\sigma^2 J_W - G_W)^2 &= \mathop{\mathbb{E}}_{\{\varepsilon_i\}} (\sigma^2 J_W - \sigma^2 J + \sigma^2 J - G_W)^2 \\
&= (\sigma^2 J_W - \sigma^2 J)^2 + \mathop{\mathbb{E}}_{\{\varepsilon_i\}} (\sigma^2 J - G_W)^2 \\
&\quad + 2 \mathop{\mathbb{E}}_{\{\varepsilon_i\}} (\sigma^2 J_W - \sigma^2 J)(\sigma^2 J - G_W). \qquad (25)
\end{aligned}$$

Eq.(19) implies

$$(\sigma^2 J_W - \sigma^2 J)^2 = O_p(n^{-3}).$$

Eqs.(19) and (10) imply

$$\begin{aligned}
2 \mathop{\mathbb{E}}_{\{\varepsilon_i\}} (\sigma^2 J_W - \sigma^2 J)(\sigma^2 J - G_W) &= 2(\sigma^2 J_W - \sigma^2 J)(\sigma^2 J - \mathop{\mathbb{E}}_{\{\varepsilon_i\}} G_W) \\
&= -2(\sigma^2 J_W - \sigma^2 J) B_W \\
&= O_p(\delta^2 n^{-\frac{5}{2}}). \qquad (26)
\end{aligned}$$

If $\delta = o_p(n^{-\frac{1}{4}})$ and the term of order $o_p(n^{-3})$ (i.e., Eq.(26)) is ignored in Eq.(25), we have

$$\begin{aligned}
\mathop{\mathbb{E}}_{\{\varepsilon_i\}} (\sigma^2 J_W - G_W)^2 &= (\sigma^2 J_W - \sigma^2 J)^2 + \mathop{\mathbb{E}}_{\{\varepsilon_i\}} (\sigma^2 J - G_W)^2 \\
&\geq \mathop{\mathbb{E}}_{\{\varepsilon_i\}} (\sigma^2 J - G_W)^2,
\end{aligned}$$

which concludes the proof. ■

References

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

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

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

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

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

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.

J. Kiefer. Optimum experimental designs. *Journal of the Royal Statistical Society, Series B*, 21: 272–304, 1959.

D. E. Knuth. *Seminumerical Algorithms*, volume 2 of *The Art of Computer Programming*. Addison-Wesley, Massachusetts, 1998.

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

F. Pukelsheim. *Optimal Design of Experiments*. John Wiley & Sons, 1993.

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

C. E. Rasmussen, R. M. Neal, G. E. Hinton, D. van Camp, M. Revow, Z. Ghahramani, R. Kustra, and R. Tibshirani. The DELVE manual, 1996. URL `http://www.cs.toronto.edu/~delve/`.

J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.

G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6:461–464, 1978.

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.

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

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.

M. Sugiyama and H. Ogawa. Active learning with model selection — Simultaneous optimization of sample points and models for trigonometric polynomial models. *IEICE Transactions on Information and Systems*, E86-D(12):2753–2763, 2003.

V. N. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, Inc., New York, 1998.

D. P. Wiens. Robust weights and designs for biased regression models: Least squares and generalized M-estimation. *Journal of Statistical Planning and Inference*, 83(2):395–412, 2000.