Mach Learn (2009) 75: 249–274
DOI 10.1007/s10994-009-5100-3

Pool-based active learning in approximate linear regression

Masashi Sugiyama · Shinichi Nakajima

Received: 30 September 2007 / Revised: 11 June 2008 / Accepted: 26 December 2008 /
Published online: 30 January 2009
Springer Science+Business Media, LLC 2009

Abstract The goal of pool-based active learning is to choose the best input points to gather output values from a 'pool' of input samples. We develop two pool-based active learning criteria for linear regression. The first criterion allows us to obtain a closed-form solution so it is computationally very efficient. However, this solution is not necessarily optimal in the single-trial generalization error analysis. The second criterion can give a better solution, but it does not have a closed-form solution and therefore some additional search strategy is needed. To cope with this problem, we propose a practical procedure which enables us to efficiently search for a better solution around the optimal solution of the first method. Simulations with toy and benchmark datasets show that the proposed active learning method compares favorably with other active learning methods as well as the baseline passive learning scheme. Furthermore, the usefulness of the proposed active learning method is also demonstrated in wafer alignment in semiconductor exposure apparatus.

Keywords Pool-based active learning · Approximate linear regression · Covariate shift · Importance-weighted least-squares · ALICE

1 Introduction

Active learning[1] (or experimental design) is a problem of optimally designing the location of training input points in supervised learning scenarios (Fedorov 1972). Choice of training

[1] In this paper, we use the term "active learning" for batch selection of training input location. However, these days, it tends to be used for a sequential choice of training input location in an interactive manner.

Editor: Lise Getoor.

M. Sugiyama (✉)
Department of Computer Science, Tokyo Institute of Technology, 2-12-1-W8-74 O-okayama,
Meguro-ku, Tokyo 152-8552, Japan
e-mail: sugi@cs.titech.ac.jp

S. Nakajima
Nikon Corporation, 201-9 Oaza-Miizugahara, Kumagaya-shi, Saitama 360-8559, Japan
e-mail: nakajima.s@nikon.co.jp

input location is particularly important when the sampling cost of output values is very high, which is often the case in the analysis of, e.g., medical data (Coomans et al. 1983), biological data (Baldi and Brunak 1998), or chemical data (Warmuth et al. 2003).

1.1 Population-based vs. pool-based active learning

Depending on the situations, active learning can be categorized into two types: *population-based* and *pool-based*.

Population-based active learning indicates the situation where we know the distribution of test input points and we are allowed to locate training input points at any desired positions (e.g., Wiens 2000; Kanamori and Shimodaira 2003; Sugiyama 2006). The goal of population-based active learning is to find the optimal training input density from which we generate training input points.

On the other hand, in pool-based active learning, the test input distribution is unknown but samples from the test input distribution are given (e.g., McCallum and Nigam 1998; Bach 2007; Kanamori 2007). The goal of pool-based active learning is to choose the best input samples from the pool of test input samples. If we have infinitely many test input samples, the pool-based problem is reduced to the population-based problem.

In this paper, we address the problem of pool-based active learning in linear regression scenarios and propose a new algorithm.

1.2 Active learning with misspecified models and covariate shift

In traditional active learning research (Fedorov 1972; Cohn et al. 1996; Fukumizu 2000), it is often assumed that the model used for function learning is *correctly specified*, i.e., it can exactly realize the learning target function. However, such an assumption may not be satisfied in reality and the violation of this assumption can cause significant performance degradation (Wiens 2000; Kanamori and Shimodaira 2003; Sugiyama 2006). For this reason, we do not assume from the beginning that our model is correct in this paper. This highly enlarges the range of application of active learning techniques.

In the active learning scenarios, the distribution of training input points is generally different from that of test input points since the location of training input points is designed by users. Such a situation is often referred to as *covariate shift* in statistics (Shimodaira 2000). Covariate shift does not matter when the model is correctly specified. However, when we deal with misspecified models, covariate shift has a significant influence—for example, *ordinary least-squares* (*OLS*) is no longer unbiased even asymptotically. Therefore, we need to explicitly take the bias caused by covariate shift into account when we work with misspecified models. A standard approach to alleviating the influence of covariate shift is to use an *importance-weighting* technique (Fishman 1996), where the term 'importance' refers to the ratio of test and training input densities. For example, in parameter learning, OLS is biased, but *Importance-Weighted Least-Squares* (*IWLS*) is asymptotically unbiased (Shimodaira 2000).

1.3 Importance estimation in pool-based active learning

In population-based active learning, importance-weighting techniques can be employed for bias reduction in a straightforward manner since the test input distribution is accessible by assumption and the training input distribution is also known since it is designed by ourselves (Wiens 2000; Kanamori and Shimodaira 2003; Sugiyama 2006). However, in pool-based active learning, the test and training input distributions may both be unknown and therefore the importance weights cannot be directly computed. A naive approach to coping with this problem is to estimate the training and test input distributions from training and test input samples. However, density estimation is known to be a hard problem particularly in high dimensional problems. Therefore, such a naive approach may not be useful in practice. This difficulty could be eased by employing recently developed methods of *direct importance estimation* (Huang et al. 2007; Bickel et al. 2007; Sugiyama et al. 2008), which allow us to obtain the importance weight without going through density estimation. However, these methods still contain some estimation error.

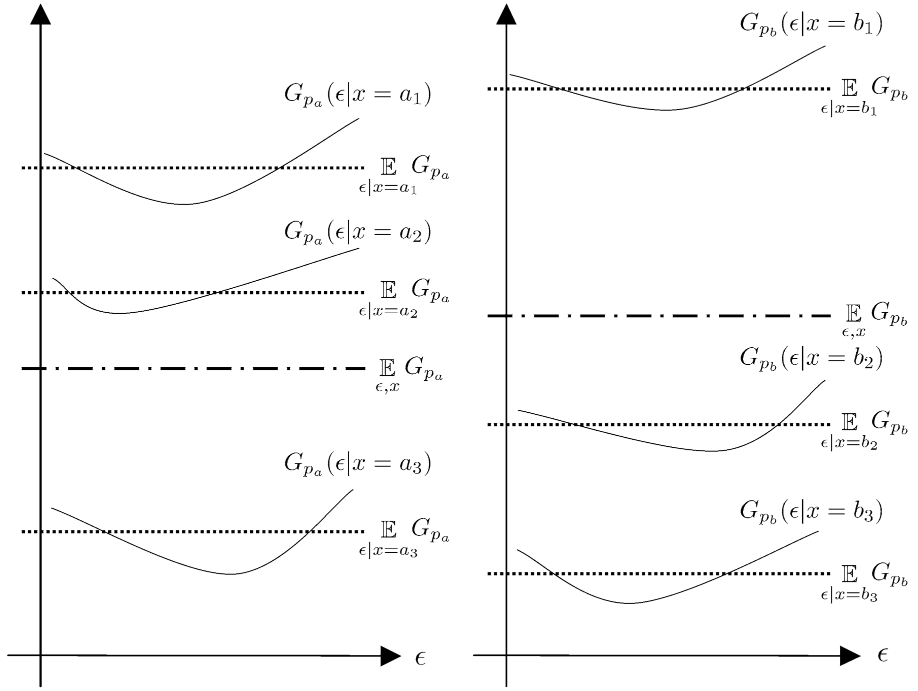
A key observation in pool-based active learning is that we choose training input points from the pool of test input points. This implies that our training input distribution is defined *over* the test input distribution, i.e., the training input distribution can be expressed as a product of the test input distribution and a *resampling bias function*. This decomposition allows us to directly compute the importance weight based on the resampling bias function, which is more accurate and computationally more efficient than the naive density estimation approach and the direct importance estimation approaches.

1.4 Single-trial analysis of generalization error

In practice, we are only given a single realization of training samples. Therefore, ideally, we want to have an estimator of the generalization error that is accurate in each *single trial*. However, we may not be able to avoid taking the expectation over the training output noise since it is not generally possible to know the realized value of noise. On the other hand, the location of the training input points is accessible by nature. Motivated by this fact, we propose to estimate the generalization error *without* taking the expectation over training input points. More specifically, we evaluate the unbiasedness of the generalization error in terms of the *conditional* expectation of training output noise given training input points (see also Sugiyama et al. 2009).

To illustrate a possible advantage of this conditional expectation approach, let us consider a simple population-based active learning scenario where only one training sample (x, y) is gathered (see Fig. 1). Suppose that the input x is drawn from a user-chosen training input distribution and y is contaminated by additive noise ϵ. The solid curves in Fig. 1(a) depict $G_{p_a}(\epsilon|x)$, 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, and a_3, respectively. The value of the generalization error for the training input 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 graph in Fig. 1(b) depicts the generalization errors for another 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 Fig. 1(a) is lower than that in Fig. 1(b). 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


$G_{p_a}(\epsilon|x = a_1)$
$\mathbb{E}_{\epsilon|x=a_1} G_{p_a}$
$G_{p_a}(\epsilon|x = a_2)$
$\mathbb{E}_{\epsilon|x=a_2} G_{p_a}$
$\mathbb{E}_{\epsilon,x} G_{p_a}$
$G_{p_a}(\epsilon|x = a_3)$
$\mathbb{E}_{\epsilon|x=a_3} G_{p_a}$
ϵ
$G_{p_b}(\epsilon|x = b_1)$
$\mathbb{E}_{\epsilon|x=b_1} G_{p_b}$
$\mathbb{E}_{\epsilon,x} G_{p_b}$
$G_{p_b}(\epsilon|x = b_2)$
$\mathbb{E}_{\epsilon|x=b_2} G_{p_b}$
$G_{p_b}(\epsilon|x = b_3)$
$\mathbb{E}_{\epsilon|x=b_3} G_{p_b}$
ϵ

(a) Generalization error for training input density p_a

(b) Generalization error for training input density p_b

Fig. 1 Schematic illustrations of the conditional-expectation and full-expectation of the generalization error

if a_3 and b_1 are realized. That is, the conditional-expectation framework may provide a finer choice of the training input density (and the training input points) than the full-expectation framework.

1.5 Contributions of this paper

We extend two population-based active learning methods proposed by Wiens (2000) and Sugiyama (2006) to pool-based scenarios. The pool-based extension of the method by Wiens (2000) allows us to obtain a closed-form solution of the best resampling bias function; thus it is computationally very efficient. However, this method is based on the full-expectation analysis of the generalization error, so the obtained solution is not necessarily optimal in terms of the single-trial generalization error. On the other hand, the pool-based extension of the method by Sugiyama (2006) can give a better solution since it is based on the conditional-expectation analysis of the generalization error. However, it does not have a closed-form solution and therefore some additional search strategy is needed.

To cope with this problem, we propose a practical procedure by combining the above two pool-based active learning methods—we use the analytic optimal solution of the full-expectation method for efficiently searching for a better solution in the conditional-expectation method. Simulations with toy and benchmark datasets show that the proposed active learning method compares favorably with other active learning methods as well as the baseline passive learning scheme. Furthermore, the proposed active learning method is shown to be also useful in wafer alignment in semiconductor exposure apparatus.

The rest of this paper is organized as follows. In Sect. 2, the complete algorithm of the proposed active learning method is described. In Sect. 3, derivation and justification of the proposed algorithm is given. In Sect. 4, the relation between the proposed and existing active learning methods is discussed. In Sect. 5, numerical results using toy and benchmark datasets are presented. In Sect. 6, the proposed method is applied to a wafer alignment problem in semiconductor exposure apparatus. Finally, in Sect. 7, concluding remarks and future prospects are given.

2 A new pool-based active learning method

In this section, we formulate the pool-based active learning problem in linear regression scenarios and describe our new algorithm. Derivation of the proposed algorithm is given in Sect. 3.

2.1 Formulation of pool-based active learning in regression

We address a regression problem of learning a real-valued function $f(\boldsymbol{x})$ defined on $\mathcal{D} \subset \mathbb{R}^d$. We are given a 'pool' of test *input* points, $\{\boldsymbol{x}_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$, which are drawn independently from an *unknown* test input distribution with density $p_{\mathrm{te}}(\boldsymbol{x})$. We assume that $p_{\mathrm{te}}(\boldsymbol{x}) > 0$ for all $\boldsymbol{x} \in \mathcal{D}$. From the pool, we are allowed to choose n_{tr} ($\ll n_{\mathrm{te}}$) input points for observing output values. Let $\{\boldsymbol{x}_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ be input points selected from the pool and $\{y_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ be corresponding output values, which are called *training samples*:

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

where $\{\epsilon_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ are i.i.d. noise with mean zero and unknown variance σ^2.

The goal of the regression task is to accurately predict the output values $\{f(\boldsymbol{x}_j^{\mathrm{te}})\}_{j=1}^{n_{\mathrm{te}}}$ at all test input points[2] $\{\boldsymbol{x}_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$. The squared loss is adopted as our error metric:

$$\frac{1}{n_{\mathrm{te}}}\sum_{j=1}^{n_{\mathrm{te}}}\left(\widehat{f}(\boldsymbol{x}_j^{\mathrm{te}}) - f(\boldsymbol{x}_j^{\mathrm{te}})\right)^2, \tag{2}$$

where $\widehat{f}(\boldsymbol{x})$ is a function learned from the training samples $\{(\boldsymbol{x}_i^{\mathrm{tr}}, y_i^{\mathrm{tr}})\}_{i=1}^{n_{\mathrm{tr}}}$.

The above formulation is summarized in Fig. 2.

2.2 Weighted least-squares for linear regression models

The following linear regression model is used for learning:

$$\widehat{f}(\boldsymbol{x}) = \sum_{\ell=1}^{t}\theta_\ell\varphi_\ell(\boldsymbol{x}), \tag{3}$$

[2] Under the assumption that $n_{\mathrm{tr}} \ll n_{\mathrm{te}}$, the difference between the prediction error at all test input points $\{\boldsymbol{x}_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$ and the remaining test input points $\{\boldsymbol{x}_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}} \backslash \{\boldsymbol{x}_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ is negligibly small. More specifically, if $n_{\mathrm{tr}} = o(\sqrt{n_{\mathrm{te}}})$, all the discussions in this paper is still valid even when the prediction error is evaluated only at the remaining test input points.

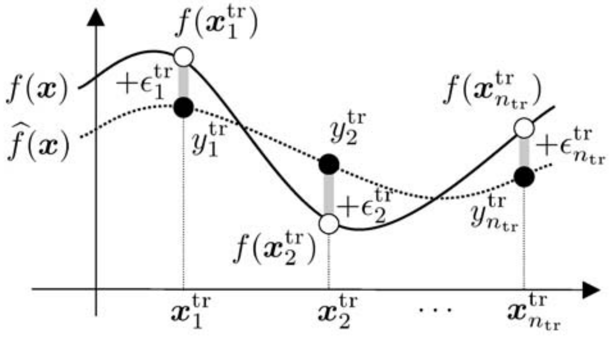

$f(\boldsymbol{x}_1^{\mathrm{tr}})$
$f(\boldsymbol{x})$
$+\epsilon_1^{\mathrm{tr}}$
$\widehat{f}(\boldsymbol{x})$
y_1^{tr}
y_2^{tr}
$f(\boldsymbol{x}_{n_{\mathrm{tr}}}^{\mathrm{tr}})$
$+\epsilon_{n_{\mathrm{tr}}}^{\mathrm{tr}}$
$+\epsilon_2^{\mathrm{tr}}$
$y_{n_{\mathrm{tr}}}^{\mathrm{tr}}$
$f(\boldsymbol{x}_2^{\mathrm{tr}})$
$\boldsymbol{x}_1^{\mathrm{tr}}$
$\boldsymbol{x}_2^{\mathrm{tr}}$
$\cdots$
$\boldsymbol{x}_{n_{\mathrm{tr}}}^{\mathrm{tr}}$

Fig. 2 Regression problem

where $\{\varphi_\ell(\boldsymbol{x})\}_{\ell=1}^t$ are fixed linearly independent basis functions. $\boldsymbol{\theta} = (\theta_1, \theta_2, \dots, \theta_t)^\top$ are parameters to be learned, where $^\top$ denotes the transpose of a vector or a matrix.

The parameter $\boldsymbol{\theta}$ of our regression model is learned by *Weighted Least-Squares* (*WLS*) with a weight function $w(\boldsymbol{x})$ (>0 for all $\boldsymbol{x} \in \mathcal{D}$), i.e.,

$$\widehat{\boldsymbol{\theta}}_{\mathrm{W}} = \underset{\boldsymbol{\theta}}{\mathrm{argmin}} \left[\sum_{i=1}^{n_{\mathrm{tr}}} w(\boldsymbol{x}_i^{\mathrm{tr}}) \left(\widehat{f}(\boldsymbol{x}_i^{\mathrm{tr}}) - y_i^{\mathrm{tr}} \right)^2 \right], \tag{4}$$

where the subscript ‘W’ denotes ‘Weighted’. Our specific choice of the weight function will be shown later. Note that the solution $\widehat{\boldsymbol{\theta}}_{\mathrm{W}}$ is invariant under constant scaling of the weight function $w(\boldsymbol{x})$ (> 0). Let $\boldsymbol{X}$ be the $n_{\mathrm{tr}} \times t$ matrix with the (i, ℓ)-th element

$$X_{i,\ell} = \varphi_\ell(\boldsymbol{x}_i^{\mathrm{tr}}). \tag{5}$$

Let $\boldsymbol{W}$ be the $n_{\mathrm{tr}} \times n_{\mathrm{tr}}$ diagonal matrix with the i-th diagonal element

$$W_{i,i} = w(\boldsymbol{x}_i^{\mathrm{tr}}). \tag{6}$$

Then $\widehat{\boldsymbol{\theta}}_{\mathrm{W}}$ is given in a closed-form as

$$\widehat{\boldsymbol{\theta}}_{\mathrm{W}} = \boldsymbol{L}_{\mathrm{W}} \boldsymbol{y}^{\mathrm{tr}}, \tag{7}$$

where

$$\boldsymbol{L}_{\mathrm{W}} = (\boldsymbol{X}^\top \boldsymbol{W} \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{W}, \tag{8}$$

$$\boldsymbol{y}^{\mathrm{tr}} = (y_1^{\mathrm{tr}}, y_2^{\mathrm{tr}}, \dots, y_{n_{\mathrm{tr}}}^{\mathrm{tr}})^\top. \tag{9}$$

2.3 Proposed active learning algorithm: P-ALICE

The goal of pool-based active learning is, from the pool of test input points $\{\boldsymbol{x}_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$, to choose the best input points $\{\boldsymbol{x}_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ for gathering output values $\{y_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ that minimizes the prediction error (2). Here, our pool-based active learning algorithm is summarized without going into the technical details; the derivation as well as its justification will be given in Sect. 3.

First, a candidate set of training input points $\{\boldsymbol{x}_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ is prepared, which is a subset of $\{\boldsymbol{x}_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$. More specifically, a *resampling bias function* $b(\boldsymbol{x})$ (>0 for all $\boldsymbol{x} \in \mathcal{D}$) is prepared and n_{tr} training input points are chosen from the pool of test input points $\{\boldsymbol{x}_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$ with probability proportional to

$$\{b(\boldsymbol{x}_j^{\mathrm{te}})\}_{j=1}^{n_{\mathrm{te}}}. \tag{10}$$

Later, we explain how a family of useful resampling bias functions is prepared. Then the 'quality' of the candidate training input points $\{\boldsymbol{x}_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ is evaluated by

$$\text{P-ALICE} = \mathrm{tr}(\widehat{\boldsymbol{U}}\boldsymbol{L}_{\mathrm{W}}\boldsymbol{L}_{\mathrm{W}}^{\top}), \tag{11}$$

where the weight function $w(\boldsymbol{x})$ included in $\boldsymbol{L}_{\mathrm{W}}$ via $\boldsymbol{W}$ is defined as[3]

$$w(\boldsymbol{x}_j^{\mathrm{te}}) = \frac{1}{b(\boldsymbol{x}_j^{\mathrm{te}})}. \tag{12}$$

$\widehat{\boldsymbol{U}}$ is the $t \times t$ matrix with the (ℓ, ℓ')-th element

$$\widehat{U}_{\ell,\ell'} = \frac{1}{n_{\mathrm{te}}}\sum_{j=1}^{n_{\mathrm{te}}} \varphi_\ell(\boldsymbol{x}_j^{\mathrm{te}})\varphi_{\ell'}(\boldsymbol{x}_j^{\mathrm{te}}). \tag{13}$$

We call the above criterion *pool-based ALICE* (*PALICE*), which is a pool-based extension of a population-based active learning criterion ALICE (*Active Learning using the Importance-weighted least-squares learning based on Conditional Expectation of the generalization error*) (Sugiyama 2006); P-ALICE is an estimator of the prediction error defined by (2), which will be detailed in Sect. 3.

Then the above evaluation is repeated for each resampling bias function in our candidate set and the best one with the smallest P-ALICE score is chosen. Once the resampling bias function and the training input points are chosen, training output values $\{y_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ are gathered at the chosen location and a linear regression model (3) is trained using WLS with the chosen weight function.

In the above procedure, the choice of the candidates of the resampling bias function $b(\boldsymbol{x})$ is arbitrary. As a heuristic, we propose using the following family of resampling bias functions parameterized by a scalar λ:

$$b_\lambda(\boldsymbol{x}) = \left(\sum_{\ell,\ell'=1}^{t} [\widehat{\boldsymbol{U}}^{-1}]_{\ell,\ell'}\varphi_\ell(\boldsymbol{x})\varphi_{\ell'}(\boldsymbol{x})\right)^{\lambda}. \tag{14}$$

The parameter λ controls the 'shape' of the training input distribution—when $\lambda = 0$, the weight is uniform over all test input samples. Thus the above choice includes *passive learning* (the training and test distributions are equivalent) as a special case. The best value of λ

[3] The expectation over a probability density $q(\boldsymbol{x})$ can be transformed into the expectation over another probability density $p(\boldsymbol{x})$ by setting the weight function $w(\boldsymbol{x})$ as the ratio of two input densities, $w(\boldsymbol{x}) = p(\boldsymbol{x})/q(\boldsymbol{x})$:

$$\int A(\boldsymbol{x})q(\boldsymbol{x})d\boldsymbol{x} = \int A(\boldsymbol{x})w(\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x},$$

which is known as the *importance sampling* technique (Fishman 1996). The situation where training and test input distributions are different is called *covariate shift* (Shimodaira 2000). Active learning naturally induces covariate shift and the bias caused by covariate shift can be compensated by the use of importance-weighted LS (see Sect. 4.1 for detail). In Sect. 3.2, we will show that the importance weight in the pool-based setting is given by the reciprocal of the resampling bias function. Note that this importance-weighting idea is a general result and its application is not limited to active learning.

```
Input: Test input points $\{\boldsymbol{x}_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$ and basis functions $\{\varphi_\ell(\boldsymbol{x})\}_{\ell=1}^{t}$.
Output: Learned parameter $\widehat{\boldsymbol{\theta}}_{\mathrm{W}}$

Compute the $t \times t$ matrix $\widehat{\boldsymbol{U}}$ with $\widehat{U}_{\ell,\ell'} = \frac{1}{n_{\mathrm{te}}} \sum_{j=1}^{n_{\mathrm{te}}} \varphi_\ell(\boldsymbol{x}_j^{\mathrm{te}}) \varphi_{\ell'}(\boldsymbol{x}_j^{\mathrm{te}})$;
For several different values of $\lambda$ (possibly around $\lambda = 1/2$)
    Compute $\{b_\lambda(\boldsymbol{x}_j^{\mathrm{te}})\}_{j=1}^{n_{\mathrm{te}}}$ with $b_\lambda(\boldsymbol{x}) = (\sum_{\ell,\ell'=1}^{t} [\widehat{\boldsymbol{U}}^{-1}]_{\ell,\ell'} \varphi_\ell(\boldsymbol{x}) \varphi_{\ell'}(\boldsymbol{x}))^{\lambda}$;
    Choose $\mathcal{X}_\lambda^{\mathrm{tr}} = \{\boldsymbol{x}_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ from $\{\boldsymbol{x}_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$ with probability proportional to $\{b_\lambda(\boldsymbol{x}_j^{\mathrm{te}})\}_{j=1}^{n_{\mathrm{te}}}$;
    Compute the $n_{\mathrm{tr}} \times t$ matrix $\boldsymbol{X}_\lambda$ with $[X_\lambda]_{i,\ell} = \varphi_\ell(\boldsymbol{x}_i^{\mathrm{tr}})$;
    Compute the $n_{\mathrm{tr}} \times n_{\mathrm{tr}}$ diagonal matrix $\boldsymbol{W}_\lambda$ with $[W_\lambda]_{i,i} = (b_\lambda(\boldsymbol{x}_i^{\mathrm{tr}}))^{-1}$;
    Compute $\boldsymbol{L}_\lambda = (\boldsymbol{X}_\lambda^\top \boldsymbol{W}_\lambda \boldsymbol{X}_\lambda)^{-1} \boldsymbol{X}_\lambda^\top \boldsymbol{W}_\lambda$;
    Compute P-ALICE$(\lambda) = \mathrm{tr}(\widehat{\boldsymbol{U}} \boldsymbol{L}_\lambda \boldsymbol{L}_\lambda^\top)$;
End
Compute $\widehat{\lambda} = \mathrm{argmin}_\lambda$ P-ALICE$(\lambda)$;
Gather training output values $\boldsymbol{y}^{\mathrm{tr}} = (y_1^{\mathrm{tr}}, y_2^{\mathrm{tr}}, \ldots, y_{n_{\mathrm{tr}}}^{\mathrm{tr}})^\top$ at $\mathcal{X}_{\widehat{\lambda}}^{\mathrm{tr}}$;
Compute $\widehat{\boldsymbol{\theta}}_{\mathrm{W}} = \boldsymbol{L}_{\widehat{\lambda}} \boldsymbol{y}^{\mathrm{tr}}$;
```

Fig. 3 Pseudo code of proposed pool-based active learning algorithm

may be searched for by simple multi-point search, i.e., the value of P-ALICE is computed for several different values of λ and the minimizer is chosen. In practice, solution search may be intensively carried out around $\lambda = 1/2$ (the reason will be explained in Sect. 3.6; an example of intensive search around $\lambda = 1/2$ is given in Sect. 5.1).

A pseudo code of the proposed pool-based active learning algorithm is described in Fig. 3.

3 Justification of proposed active learning algorithm

In this section, we explain how we came up with the active learning algorithm described in Sect. 2.3.

3.1 Overview of this section

The proposed P-ALICE criterion (11) is an extention of a population-based active learning criterion called ALICE (*Active Learning using the Importance-weighted least-squares learning based on Conditional Expectation of the generalization error*)[4] (Sugiyama 2006) to pool-based scenarios. Our choice of candidates of the resampling bias function (14) is motivated by a pool-based extension of another population-based active learning method which we call *Full-expectation Variance-only active learning for WLS* (FV$_{\mathrm{W}}$) (Wiens 2000).

We review ALICE in Sect. 3.2 and extend it to the pool-based scenarios in Sect. 3.3. Then we review FV$_{\mathrm{W}}$ in Sect. 3.4 and extend it to the pool-based scenarios in Sect. 3.5. Finally, in Sect. 3.6, P-ALICE and P-FV$_{\mathrm{W}}$ are combined and the proposed active learning algorithm is obtained.

[4] ALICE corresponds to *Conditional-expectation Variance-only active learning for WLS* (CV$_{\mathrm{W}}$), if we express the name consistent with other methods.

3.2 Population-based active learning criterion: ALICE

Here we review a population-based active learning criterion ALICE.

In the population-based framework, the test input density $p_{\mathrm{te}}(\boldsymbol{x})$ is given (e.g., Fukumizu 2000; Wiens 2000; Kanamori and Shimodaira 2003; Sugiyama 2006). The goal is to determine the best training input density $p_{\mathrm{tr}}(\boldsymbol{x})$ from which training input points $\{\boldsymbol{x}_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ are drawn.

The aim of the regression task in the population-based framework is to accurately predict the output values for all test input samples drawn from $p_{\mathrm{te}}(\boldsymbol{x})$. Thus the error metric (often called the *generalization error*) is

$$G' = \int \left(\widehat{f}(\boldsymbol{x}^{\mathrm{te}}) - f(\boldsymbol{x}^{\mathrm{te}})\right)^2 p_{\mathrm{te}}(\boldsymbol{x}^{\mathrm{te}})d\boldsymbol{x}^{\mathrm{te}}. \tag{15}$$

Suppose the regression model (3) *approximately*[5] includes the learning target function $f(\boldsymbol{x})$, i.e., for a scalar δ such that $|\delta|$ is small, $f(\boldsymbol{x})$ is expressed as

$$f(\boldsymbol{x}) = g(\boldsymbol{x}) + \delta r(\boldsymbol{x}), \tag{16}$$

where $g(\boldsymbol{x})$ is the optimal approximation to $f(\boldsymbol{x})$ by the model (3):

$$g(\boldsymbol{x}) = \sum_{\ell=1}^{t} \theta_\ell^* \varphi_\ell(\boldsymbol{x}). \tag{17}$$

$\boldsymbol{\theta}^* = (\theta_1^*, \theta_2^*, \ldots, \theta_t^*)^\top$ is the unknown optimal parameter defined by

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\mathrm{argmin}}\, G'. \tag{18}$$

$\delta r(\boldsymbol{x})$ in (16) is the residual function, which is orthogonal to $\{\varphi_\ell(\boldsymbol{x})\}_{\ell=1}^{t}$ under $p_{\mathrm{te}}(\boldsymbol{x})$ (see Fig. 4):

$$\int r(\boldsymbol{x}^{\mathrm{te}})\varphi_\ell(\boldsymbol{x}^{\mathrm{te}})p_{\mathrm{te}}(\boldsymbol{x}^{\mathrm{te}})d\boldsymbol{x}^{\mathrm{te}} = 0 \quad \text{for } \ell = 1, 2, \ldots, t. \tag{19}$$

The function $r(\boldsymbol{x})$ governs the nature of the model error, while δ is the possible magnitude of this error. In order to separate these two factors, the following normalization condition on $r(\boldsymbol{x})$ is further imposed:

$$\int \left(r(\boldsymbol{x}^{\mathrm{te}})\right)^2 p_{\mathrm{te}}(\boldsymbol{x}^{\mathrm{te}})d\boldsymbol{x}^{\mathrm{te}} = 1. \tag{20}$$

Let $\mathbb{E}_{\{\epsilon_i\}_{i=1}^{n_{\mathrm{tr}}}}$ be the expectation over the noise $\{\epsilon_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$. Then, the generalization error expected over the training output noise can be decomposed into[6] the (squared) *bias* term B,

[5] In traditional active learning literature (Fedorov 1972; Cohn et al. 1996; Fukumizu 2000), the model is often assumed to be correctly specified, i.e., the target function $f(\boldsymbol{x})$ can be realized by the model (3). However, this may not be satisfied in practice and these methods are shown to perform poorly when model correctness is not fulfilled (e.g., Wiens 2000; Kanamori and Shimodaira 2003; Sugiyama 2006). On the other hand, some domain-specific knowledge is often available and it may be possible to construct a 'good' model, which is not exactly correct, but approximately correct. This is the situation we are addressing here. When the model is heavily misspecified, it is necessary to perform model selection, which is discussed in Sugiyama and Rubens (2008); see also Sect. 7.

[6] Sometimes $B + \delta^2$ is referred to as the bias, but they are treated separately here since B is reducible while δ^2 is constant for a fixed model.

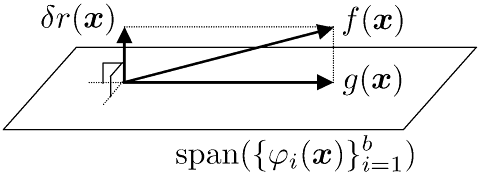

$\delta r(\boldsymbol{x})$
$f(\boldsymbol{x})$
$g(\boldsymbol{x})$
$\mathrm{span}(\{\varphi_i(\boldsymbol{x})\}_{i=1}^{b})$

Fig. 4 Orthogonal decomposition of $f(\boldsymbol{x}^{\mathrm{tr}})$

the *variance* term V, and the model error δ^2:

$$\mathop{\mathbb{E}}_{\{\epsilon_i\}_{i=1}^{n_{\mathrm{tr}}}} G' = B + V + \delta^2, \tag{21}$$

where

$$B = \int \left(\mathop{\mathbb{E}}_{\{\epsilon_i\}_{i=1}^{n_{\mathrm{tr}}}} \widehat{f}(\boldsymbol{x}^{\mathrm{te}}) - g(\boldsymbol{x}^{\mathrm{te}}) \right)^2 p_{\mathrm{te}}(\boldsymbol{x}^{\mathrm{te}}) d\boldsymbol{x}^{\mathrm{te}}, \tag{22}$$

$$V = \int \mathop{\mathbb{E}}_{\{\epsilon_i\}_{i=1}^{n_{\mathrm{tr}}}} \left(\widehat{f}(\boldsymbol{x}^{\mathrm{te}}) - \mathop{\mathbb{E}}_{\{\epsilon_i\}_{i=1}^{n_{\mathrm{tr}}}} \widehat{f}(\boldsymbol{x}^{\mathrm{te}}) \right)^2 p_{\mathrm{te}}(\boldsymbol{x}^{\mathrm{te}}) d\boldsymbol{x}^{\mathrm{te}}. \tag{23}$$

Since δ is constant which depends neither on $p_{\mathrm{tr}}(\boldsymbol{x})$ nor $\{\boldsymbol{x}_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$, δ^2 is subtracted from G' and define it by G.

$$G = G' - \delta^2. \tag{24}$$

For parameter learning, *importance-weighted least-squares* (IWLS) is used (Shimodaira 2000), i.e., (4) with the weight function $w(\boldsymbol{x})$ being the ratio of densities called the *importance ratio*:

$$w(\boldsymbol{x}) = \frac{p_{\mathrm{te}}(\boldsymbol{x})}{p_{\mathrm{tr}}(\boldsymbol{x})}. \tag{25}$$

The solution $\widehat{\boldsymbol{\theta}}_{\mathrm{W}}$ is given by (7).

Let G_{W}, B_{W}, and V_{W} be G, B, and V for the learned function obtained by IWLS, respectively. Let $\boldsymbol{U}$ be the $t \times t$ matrix with the (ℓ, ℓ')-th element

$$U_{\ell,\ell'} = \int \varphi_\ell(\boldsymbol{x}^{\mathrm{te}}) \varphi_{\ell'}(\boldsymbol{x}^{\mathrm{te}}) p_{\mathrm{te}}(\boldsymbol{x}^{\mathrm{te}}) d\boldsymbol{x}^{\mathrm{te}}. \tag{26}$$

Then, for IWLS with an approximately correct model, B and V are expressed as follows (Sugiyama 2006):

$$B_{\mathrm{W}} = \mathcal{O}_p(\delta^2 n_{\mathrm{tr}}^{-1}), \tag{27}$$

$$V_{\mathrm{W}} = \sigma^2 \mathrm{tr}(\boldsymbol{U} \boldsymbol{L}_{\mathrm{W}} \boldsymbol{L}_{\mathrm{W}}^{\top}) = \mathcal{O}_p(n_{\mathrm{tr}}^{-1}). \tag{28}$$

Note that the asymptotic order in the above equations is in probability since random variables $\{\boldsymbol{x}_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ are included. The above equations imply that if[7] $\delta = o_p(1)$,

$$\mathop{\mathbb{E}}_{\{\epsilon_i\}_{i=1}^{n_{\mathrm{tr}}}} G_{\mathrm{W}} = \sigma^2 \mathrm{tr}(\boldsymbol{U}\boldsymbol{L}_{\mathrm{W}}\boldsymbol{L}_{\mathrm{W}}^{\top}) + o_p(n_{\mathrm{tr}}^{-1}). \tag{29}$$

The active learning criterion ALICE is motivated by this asymptotic form, i.e., ALICE chooses the training input density $p_{\mathrm{tr}}(\boldsymbol{x})$ from the set $\mathcal{P}$ of all strictly positive probability densities[8] as

$$p_{\mathrm{tr}}^{\mathrm{ALICE}} = \mathop{\mathrm{argmin}}_{p_{\mathrm{tr}} \in \mathcal{P}} \mathrm{ALICE}, \tag{30}$$

where

$$\mathrm{ALICE} = \mathrm{tr}(\boldsymbol{U}\boldsymbol{L}_{\mathrm{W}}\boldsymbol{L}_{\mathrm{W}}^{\top}). \tag{31}$$

Practically, $\mathcal{P}$ may be replaced by a finite set $\widehat{\mathcal{P}}$ of strictly positive probability densities and choose the one that minimizes ALICE from the set $\widehat{\mathcal{P}}$.

3.3 Extension of ALICE to pool-based scenarios: P-ALICE

Our basic idea of P-ALICE is to extend the population-based ALICE method to the pool-based scenario, where $p_{\mathrm{te}}(\boldsymbol{x})$ is unknown, but a pool of test input samples $\{\boldsymbol{x}_i^{\mathrm{te}}\}_{i=1}^{n_{\mathrm{te}}}$ drawn independently from $p_{\mathrm{te}}(\boldsymbol{x})$ is given. Under the pool-based setting, the following two quantities included in ALICE are inaccessible:

(A) The expectation over $p_{\mathrm{te}}(\boldsymbol{x})$ contained in $\boldsymbol{U}$.
(B) The importance ratio $p_{\mathrm{te}}(\boldsymbol{x})/p_{\mathrm{tr}}(\boldsymbol{x})$ at training input points $\{\boldsymbol{x}_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ contained in $\boldsymbol{L}_{\mathrm{W}}$ through $\boldsymbol{W}$.

Regarding (A), the expectation over $p_{\mathrm{te}}(\boldsymbol{x})$ may be approximated by the expectation over test input samples $\{\boldsymbol{x}_i^{\mathrm{te}}\}_{i=1}^{n_{\mathrm{te}}}$, which is known to be *consistent*. However, approximating (B) is not straightforward (as explained in Sect. 1.3).

In pool-based active learning, training input points are chosen from the pool of test input points following a resampling bias function $b(\boldsymbol{x})$. This implies that our training input distribution is defined *over* the test input distribution, i.e., the training input distribution is expressed as a product of the test input distribution and a resampling bias function $b(\boldsymbol{x})$ (cf. Kanamori 2007):

$$p_{\mathrm{tr}}(\boldsymbol{x}_j^{\mathrm{te}}) \propto p_{\mathrm{te}}(\boldsymbol{x}_j^{\mathrm{te}}) b(\boldsymbol{x}_j^{\mathrm{te}}). \tag{32}$$

This immediately shows that the importance weight $w(\boldsymbol{x}_j^{\mathrm{te}})$ is given by

$$w(\boldsymbol{x}_j^{\mathrm{te}}) \propto \frac{1}{b(\boldsymbol{x}_j^{\mathrm{te}})}. \tag{33}$$

[7] Since δ is the model error which is a constant, the expression $\delta = o_p(1)$ is not achievable in reality. However, such an assumption seems common in the analysis of approximately correct models and this roughly means δ is small.

[8] More precisely, ALICE depends not only on the training input density $p_{\mathrm{tr}}(\boldsymbol{x})$, but also the realized values $\{\boldsymbol{x}_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ of the input points. See Sect. 7 for some additional discussions on this issue.

Note that the scaling factor of $w(\boldsymbol{x})$ is irrelevant in IWLS (cf. (4)). Equation (33) is more accurate and computationally more efficient than the naive density estimation approach and the direct importance estimation approaches. Consequently, we obtain the P-ALICE criterion (11).

3.4 Population-based active learning criterion: $\mathrm{FV_W}$

Next, we show how we came up with the candidate set of resampling bias functions given in (14). Our choice is based on a population-based active learning method proposed by Wiens (2000). First, we consider the population-based setting and briefly review this method.

For IWLS, Kanamori and Shimodaira (2003) proved that the generalization error expected over training input points $\{\boldsymbol{x}_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ and training output noise $\{\epsilon_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ is asymptotically expressed as

$$\mathop{\mathbb{E}}_{\{\boldsymbol{x}_i\}_{i=1}^{n_{\mathrm{tr}}}} \mathop{\mathbb{E}}_{\{\epsilon_i\}_{i=1}^{n_{\mathrm{tr}}}} G_{\mathrm{W}} = \frac{1}{n_{\mathrm{tr}}}\mathrm{tr}(\boldsymbol{U}^{-1}\boldsymbol{S}) + \frac{\sigma^2}{n_{\mathrm{tr}}}\mathrm{tr}(\boldsymbol{U}^{-1}\boldsymbol{T}) + \mathcal{O}({n_{\mathrm{tr}}}^{-\frac{3}{2}}), \tag{34}$$

where $\mathbb{E}_{\{\boldsymbol{x}_i\}_{i=1}^{n_{\mathrm{tr}}}}$ is the expectation over training input points $\{\boldsymbol{x}_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$. $\boldsymbol{S}$ and $\boldsymbol{T}$ are the $t \times t$ matrices with the (ℓ, ℓ')-th elements

$$S_{\ell,\ell'} = \delta^2 \int \varphi_\ell(\boldsymbol{x}^{\mathrm{te}})\varphi_{\ell'}(\boldsymbol{x}^{\mathrm{te}}) \left(r(\boldsymbol{x}^{\mathrm{te}})\right)^2 w(\boldsymbol{x}^{\mathrm{te}}) p_{\mathrm{te}}(\boldsymbol{x}^{\mathrm{te}}) d\boldsymbol{x}^{\mathrm{te}}, \tag{35}$$

$$T_{\ell,\ell'} = \int \varphi_\ell(\boldsymbol{x}^{\mathrm{te}})\varphi_{\ell'}(\boldsymbol{x}^{\mathrm{te}}) w(\boldsymbol{x}^{\mathrm{te}}) p_{\mathrm{te}}(\boldsymbol{x}^{\mathrm{te}}) d\boldsymbol{x}^{\mathrm{te}}, \tag{36}$$

where $w(\boldsymbol{x})$ above is the importance ratio (25). Note that $\frac{1}{n_{\mathrm{tr}}}\mathrm{tr}(\boldsymbol{U}^{-1}\boldsymbol{S})$ corresponds to the squared bias while $\frac{\sigma^2}{n_{\mathrm{tr}}}\mathrm{tr}(\boldsymbol{U}^{-1}\boldsymbol{T})$ corresponds to the variance.

It can be shown (Kanamori and Shimodaira 2003; Sugiyama 2006) that if $\delta = o(1)$,

$$\mathop{\mathbb{E}}_{\{\boldsymbol{x}_i\}_{i=1}^{n_{\mathrm{tr}}}} \mathop{\mathbb{E}}_{\{\epsilon_i\}_{i=1}^{n_{\mathrm{tr}}}} G_{\mathrm{W}} = \frac{\sigma^2}{n_{\mathrm{tr}}}\mathrm{tr}(\boldsymbol{U}^{-1}\boldsymbol{T}) + o({n_{\mathrm{tr}}}^{-1}). \tag{37}$$

Based on this asymptotic form, a population-based active learning criterion, which we refer to as *Full-expectation Variance-only active learning for WLS* ($\mathrm{FV_W}$), is given as follows (Wiens 2000):

$$p_{\mathrm{tr}}^{\mathrm{FV_W}} = \mathop{\mathrm{argmin}}_{p_{\mathrm{tr}} \in \mathcal{P}} \mathrm{FV_W}, \tag{38}$$

where

$$\mathrm{FV_W} = \frac{1}{n_{\mathrm{tr}}}\mathrm{tr}(\boldsymbol{U}^{-1}\boldsymbol{T}). \tag{39}$$

A notable feature of $\mathrm{FV_W}$ is that the optimal training input density $p_{\mathrm{tr}}^{\mathrm{FV_W}}(\boldsymbol{x})$ can be obtained in a closed-form (Wiens 2000; Kanamori 2007):

$$p_{\mathrm{tr}}^{\mathrm{FV_W}}(\boldsymbol{x}) \propto p_{\mathrm{te}}(\boldsymbol{x}) b_{\mathrm{FV_W}}(\boldsymbol{x}), \tag{40}$$

where

$$b_{\mathrm{FV_W}}(\boldsymbol{x}) = \left(\sum_{\ell,\ell'=1}^{t} [\boldsymbol{U}^{-1}]_{\ell,\ell'} \varphi_\ell(\boldsymbol{x}) \varphi_{\ell'}(\boldsymbol{x}) \right)^{\frac{1}{2}}. \tag{41}$$

Note that (40) implies that the importance ratio for the optimal training input density $p_{\mathrm{tr}}^{\mathrm{FV_W}}(\boldsymbol{x})$ is given by

$$w_{\mathrm{FV_W}}(\boldsymbol{x}) \propto \frac{1}{b_{\mathrm{FV_W}}(\boldsymbol{x})}. \tag{42}$$

3.5 Extension of $\mathrm{FV_W}$ to pool-based scenarios: P-$\mathrm{FV_W}$

If the values of the function $b_{\mathrm{FV_W}}(\boldsymbol{x})$ at the test input points $\{\boldsymbol{x}_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$ are available, they can be used as a resampling bias function in pool-based active learning. However, since $\boldsymbol{U}$ is unknown in the pool-based scenario, it is not possible to directly compute the values of $b_{\mathrm{FV_W}}(\boldsymbol{x})$ at the test input points $\{\boldsymbol{x}_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$. To cope with this problem, we propose replacing $\boldsymbol{U}$ with an empirical estimate $\widehat{\boldsymbol{U}}$. Then, the resampling bias function $\{b_{\mathrm{P\text{-}FV_W}}(\boldsymbol{x}_j^{\mathrm{te}})\}_{j=1}^{n_{\mathrm{te}}}$ is given by

$$b_{\mathrm{P\text{-}FV_W}}(\boldsymbol{x}_j^{\mathrm{te}}) = \left(\sum_{\ell,\ell'=1}^{t} [\widehat{\boldsymbol{U}}^{-1}]_{\ell,\ell'} \varphi_\ell(\boldsymbol{x}_j^{\mathrm{te}}) \varphi_{\ell'}(\boldsymbol{x}_j^{\mathrm{te}}) \right)^{\frac{1}{2}}. \tag{43}$$

The importance weight is given as

$$w_{\mathrm{P\text{-}FV_W}}(\boldsymbol{x}_j^{\mathrm{te}}) \propto \frac{1}{b_{\mathrm{P\text{-}FV_W}}(\boldsymbol{x}_j^{\mathrm{te}})}. \tag{44}$$

3.6 Combining P-ALICE and P-$\mathrm{FV_W}$

It was shown that P-$\mathrm{FV_W}$ has a closed-form solution of the optimal resampling bias function. This directly suggests to use $b_{\mathrm{P\text{-}FV_W}}(\boldsymbol{x}_j^{\mathrm{te}})$ for active learning. Nevertheless, we argue that it is possible to further improve the solution.

The point of our argument is the way the generalization error is analyzed—the optimality of $\mathrm{FV_W}$ is in terms of the expectation over *both* training input points $\{\boldsymbol{x}_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ and training output noise $\{\epsilon_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ (see (34)), while ALICE is optimal in terms of the *conditional* expectation over training output noise $\{\epsilon_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ given $\{\boldsymbol{x}_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$. The former is called the *full-expectation* (or data-independent) analysis while the latter is called the *conditional-expectation* (or input-dependent) analysis (Sugiyama et al. 2009).

What we really want to evaluate in reality is the *single-trial* generalization error, i.e., the generalization error where both $\{\boldsymbol{x}_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ and $\{\epsilon_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ are given and fixed. However, the single-trial generalization error cannot be directly evaluated since the noise $\{\epsilon_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ is not accessible in practice. On the other hand, the training input points $\{\boldsymbol{x}_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ are known and accessible in the current setting. The idea of the conditional-expectation approach is to make use of the information provided by the realized input points $\{\boldsymbol{x}_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$. It was shown that the conditional-expectation approach is provably more accurate in the single-trial analysis than the full-expectation approach (Sugiyama 2006), which is explained below.

ALICE and $\mathrm{FV_W}$ are both variance estimators; the difference is that ALICE is an estimator of conditional variance expected over $\{\epsilon_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ given $\{\boldsymbol{x}_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$, while $\mathrm{FV_W}$ is an estimator of full variance expected over both $\{\epsilon_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ and $\{\boldsymbol{x}_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$. ALICE (31) and $\mathrm{FV_W}$ (39) are related to each other by

$$\mathrm{ALICE} = \mathrm{FV_W} + \mathcal{O}_p(n_{\mathrm{tr}}^{-\frac{3}{2}}), \tag{45}$$

implying that they are actually equivalent asymptotically. However, they are different in the order of n_{tr}^{-1}; indeed, if $\delta = o_p(n_{\mathrm{tr}}^{-\frac{1}{4}})$ and terms of $o_p(n_{\mathrm{tr}}^{-3})$ are ignored, the following inequality holds (see Sugiyama 2006, for its proof):

$$\mathop{\mathbb{E}}_{\{\epsilon_i\}_{i=1}^{n_{\mathrm{tr}}}} (\sigma^2 \mathrm{FV_W} - G_{\mathrm{W}})^2 \geq \mathop{\mathbb{E}}_{\{\epsilon_i\}_{i=1}^{n_{\mathrm{tr}}}} (\sigma^2 \mathrm{ALICE} - G_{\mathrm{W}})^2. \tag{46}$$

This implies that σ^2ALICE is asymptotically a more accurate estimator of the single-trial generalization error G_{W} than $\sigma^2\mathrm{FV_W}$.

This analysis suggests that using P-ALICE is more suitable than P-$\mathrm{FV_W}$. However, a drawback of P-ALICE is that a closed-form solution is not available—thus, candidates of training input samples need to be prepared and the best solution should be searched for from the candidates. To ease this problem, our heuristic is to use the closed-form solution of P-$\mathrm{FV_W}$ as a 'base' candidate and search for a better solution around the vicinity of the P-$\mathrm{FV_W}$ solution. More specifically, a family of resampling bias functions (14) is considered, which is parameterized by λ. This family consists of the optimal solution of P-$\mathrm{FV_W}$ ($\lambda = 1/2$) and its variants ($\lambda \neq 1/2$); passive learning is also included as a special case ($\lambda = 0$) in this family. We note that the way the resampling bias function is parameterized in (14) is just a heuristic; alternative strategies may be used for parameterizing resampling bias functions. Using a richer function family will improve the search performance, but this increases the computation time in turn. The current heuristic is very simple and contain only one parameter λ, but we experimentally show in Sect. 5 that this simple heuristic works well.

The extensive experimental results in Sect. 5 show that an additional search using P-ALICE tends to improve the active learning performance over P-$\mathrm{FV_W}$.

4 Relation to existing methods

In this section, the proposed active learning method is qualitatively compared with existing methods.

4.1 Conditional-expectation variance-only active learning for OLS: $\mathrm{CV_O}$

Let us begin with population-based scenarios. A traditional way to learn the parameters in the regression model (3) is *Ordinary Last-Squares* (*OLS*), i.e., the parameter vector $\boldsymbol{\theta}$ is determined as follows.

$$\widehat{\boldsymbol{\theta}}_{\mathrm{O}} = \mathop{\mathrm{argmin}}_{\boldsymbol{\theta}} \left[\sum_{i=1}^{n_{\mathrm{tr}}} \left(\widehat{f}(\boldsymbol{x}_i^{\mathrm{tr}}) - y_i^{\mathrm{tr}} \right)^2 \right], \tag{47}$$

where the subscript 'O' denotes 'Ordinary'. $\widehat{\boldsymbol{\theta}}_{\mathrm{O}}$ is analytically given by

$$\widehat{\boldsymbol{\theta}}_{\mathrm{O}} = \boldsymbol{L}_{\mathrm{O}} \boldsymbol{y}^{\mathrm{tr}}, \tag{48}$$

where

$$\boldsymbol{L}_{\mathrm{O}} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1}\boldsymbol{X}^\top. \tag{49}$$

Let G_{O}, B_{O}, and V_{O} be G, B, and V for the learned function obtained by OLS, respectively. For an approximately correct model, B_{O} and V_{O} are expressed as follows (e.g., Sugiyama 2006):

$$B_{\mathrm{O}} = \mathcal{O}(\delta^2), \tag{50}$$

$$V_{\mathrm{O}} = \sigma^2 \mathrm{tr}(\boldsymbol{U}\boldsymbol{L}_{\mathrm{O}}\boldsymbol{L}_{\mathrm{O}}^\top) = \mathcal{O}_p(n_{\mathrm{tr}}^{-1}). \tag{51}$$

Motivated by these asymptotic forms, a population-based active learning method, which we refer to as $\mathrm{CV_O}$ (*Conditional-expectation Variance-only active learning for OLS*), optimizes the training input density by the following criterion (Fedorov 1972; Cohn et al. 1996; Fukumizu 2000).

$$p_{\mathrm{tr}}^{\mathrm{CV_O}} = \mathop{\mathrm{argmin}}_{p_{\mathrm{tr}} \in \mathcal{P}} \mathrm{CV_O}, \tag{52}$$

where

$$\mathrm{CV_O} = \mathrm{tr}(\boldsymbol{U}\boldsymbol{L}_{\mathrm{O}}\boldsymbol{L}_{\mathrm{O}}^\top). \tag{53}$$

As shown in Sugiyama (2006), if $\delta = o_p(n_{\mathrm{tr}}^{-\frac{1}{2}})$,

$$\mathop{\mathbb{E}}_{\{\epsilon_i\}_{i=1}^{n_{\mathrm{tr}}}} G_{\mathrm{O}} = \sigma^2 V_{\mathrm{O}} + o_p(n_{\mathrm{tr}}^{-1}). \tag{54}$$

Thus, $\mathrm{CV_O}$ requires $\delta = o_p(n_{\mathrm{tr}}^{-\frac{1}{2}})$ for its valid use. On the other hand, ALICE requires $\delta = o_p(1)$, which is weaker than $\mathrm{CV_O}$. Therefore, ALICE has a wider range of applications than $\mathrm{CV_O}$. This difference comes from the fact that in active learning, the training input points and test input points are generally drawn from different distributions, which is often referred to as *covariate shift* (Shimodaira 2000). Under covariate shift, if the model is misspecified, OLS is not unbiased even asymptotically; instead, IWLS is asymptotically unbiased. Asymptotic unbiasedness of IWLS would be intuitively understood by the following identity (Fishman 1996):

$$\int \left(\widehat{f}(\boldsymbol{x}^{\mathrm{te}}) - f(\boldsymbol{x}^{\mathrm{te}})\right)^2 p_{\mathrm{te}}(\boldsymbol{x}^{\mathrm{te}}) d\boldsymbol{x}^{\mathrm{te}} = \int \left(\widehat{f}(\boldsymbol{x}^{\mathrm{tr}}) - f(\boldsymbol{x}^{\mathrm{tr}})\right)^2 w(\boldsymbol{x}^{\mathrm{tr}}) p_{\mathrm{tr}}(\boldsymbol{x}^{\mathrm{tr}}) d\boldsymbol{x}^{\mathrm{tr}}, \tag{55}$$

where $w(\boldsymbol{x})$ above is the importance ratio (25).

$\mathrm{CV_O}$ can be immediately extended to a pool-based method just by replacing $\boldsymbol{U}$ with $\widehat{\boldsymbol{U}}$, i.e.,

$$\text{P-CV}_{\mathrm{O}} = \mathrm{tr}(\widehat{\boldsymbol{U}}\boldsymbol{L}_{\mathrm{O}}\boldsymbol{L}_{\mathrm{O}}^\top). \tag{56}$$

Note that $\mathrm{CV_O}$ is often referred to as the *Q-optimal design* (Fedorov 1972). The *A-optimal design* and *D-optimal design* are related active learning criteria which minimize the trace and determinant of the covariance matrix $\boldsymbol{L}_{\mathrm{O}}\boldsymbol{L}_{\mathrm{O}}^\top$, respectively. Although A-optimality and D-optimality are different from Q-optimality, they all share the common drawback for misspecified models, i.e., the bias B can be large.

4.2 Full-expectation bias-and-variance active learning for OLS and WLS: $\mathrm{FBV}_{\mathrm{OW}}$

Let us again begin with population-based scenarios. Let $\boldsymbol{H}$ be the $t \times t$ matrix defined by

$$\boldsymbol{H} = \boldsymbol{S} + \sigma^2 \boldsymbol{T}, \tag{57}$$

where $\boldsymbol{S}$ and $\boldsymbol{T}$ are defined in (35) and (36), respectively. Then (34) is expressed as

$$\mathop{\mathbb{E}}_{\{\boldsymbol{x}_i\}_{i=1}^{n_{\mathrm{tr}}}} \mathop{\mathbb{E}}_{\{\epsilon_i\}_{i=1}^{n_{\mathrm{tr}}}} G_{\mathrm{W}} = \frac{1}{n_{\mathrm{tr}}} \mathrm{tr}(\boldsymbol{U}^{-1}\boldsymbol{H}) + \mathcal{O}(n_{\mathrm{tr}}^{-\frac{3}{2}}). \tag{58}$$

Kanamori and Shimodaira (2003) developed a method to approximate $\boldsymbol{H}$ by a two-stage sampling scheme: the training samples gathered in the first stage are used for estimating $\boldsymbol{H}$ and the distribution of the remaining training input points is optimized based on the estimated $\boldsymbol{H}$ in the second stage. A more detailed description is given below.

First, $\widetilde{n}_{\mathrm{tr}}$ ($\leq n_{\mathrm{tr}}$) *initial* training input points $\{\widetilde{\boldsymbol{x}}_i^{\mathrm{tr}}\}_{i=1}^{\widetilde{n}_{\mathrm{tr}}}$ are created independently following the test input distribution with density $p_{\mathrm{te}}(\boldsymbol{x})$, and corresponding output values $\{\widetilde{y}_i^{\mathrm{tr}}\}_{i=1}^{\widetilde{n}_{\mathrm{tr}}}$ are observed. Let $\widetilde{\boldsymbol{D}}$ and $\widetilde{\boldsymbol{Q}}$ be the $\widetilde{n}_{\mathrm{tr}} \times \widetilde{n}_{\mathrm{tr}}$ diagonal matrices with the i-th diagonal elements

$$\widetilde{D}_{i,i} = \frac{p_{\mathrm{te}}(\widetilde{\boldsymbol{x}}_i^{\mathrm{tr}})}{p_{\mathrm{tr}}(\widetilde{\boldsymbol{x}}_i^{\mathrm{tr}})}, \tag{59}$$

$$\widetilde{Q}_{i,i} = [\widetilde{\boldsymbol{y}}^{\mathrm{tr}} - \widetilde{\boldsymbol{X}}(\widetilde{\boldsymbol{X}}^\top \widetilde{\boldsymbol{X}})^{-1}\widetilde{\boldsymbol{X}}^\top \widetilde{\boldsymbol{y}}^{\mathrm{tr}}]_i, \tag{60}$$

where $\widetilde{\boldsymbol{X}}$ is the $\widetilde{n}_{\mathrm{tr}} \times t$ matrix with the (i, ℓ)-th element

$$\widetilde{X}_{i,\ell} = \varphi_\ell(\widetilde{\boldsymbol{x}}_i^{\mathrm{tr}}), \tag{61}$$

and

$$\widetilde{\boldsymbol{y}}^{\mathrm{tr}} = (\widetilde{y}_1^{\mathrm{tr}}, \widetilde{y}_2^{\mathrm{tr}}, \ldots, \widetilde{y}_{\widetilde{n}_{\mathrm{tr}}}^{\mathrm{tr}})^\top. \tag{62}$$

Then an approximation $\widetilde{\boldsymbol{H}}$ of the unknown matrix $\boldsymbol{H}$ in (58) is given by

$$\widetilde{\boldsymbol{H}} = \frac{1}{\widetilde{n}_{\mathrm{tr}}} \widetilde{\boldsymbol{X}}^\top \widetilde{\boldsymbol{D}} \widetilde{\boldsymbol{Q}}^2 \widetilde{\boldsymbol{X}}. \tag{63}$$

Based on this approximation, a population-based active learning criterion, which we refer to as *Full-expectation Bias-and-Variance active learning for OLS and WLS method* ($\mathrm{FBV}_{\mathrm{OW}}$), is given as

$$p_{\mathrm{tr}}^{\mathrm{FBV}_{\mathrm{OW}}} = \mathop{\mathrm{argmin}}_{p_{\mathrm{tr}} \in \mathcal{P}} \mathrm{FBV}_{\mathrm{OW}}, \tag{64}$$

where

$$\mathrm{FBV}_{\mathrm{OW}} = \frac{1}{n_{\mathrm{tr}}} \mathrm{tr}(\widetilde{\boldsymbol{U}}^{-1}\widetilde{\boldsymbol{H}}), \tag{65}$$

$$\widetilde{\boldsymbol{U}} = \frac{1}{\widetilde{n}_{\mathrm{tr}}} \widetilde{\boldsymbol{X}}^\top \widetilde{\boldsymbol{X}}. \tag{66}$$

Note that in (65), $\boldsymbol{U}$ is replaced by its consistent estimator $\widetilde{\boldsymbol{U}}$. However, this replacement may not be necessary when the test input density $p_{\mathrm{te}}(\boldsymbol{x})$ is known (Sugiyama 2006).

After determining the optimal density $p_{\mathrm{tr}}^{\mathrm{FBV}_{\mathrm{OW}}}(\boldsymbol{x})$, the remaining $(n_{\mathrm{tr}} - \widetilde{n}_{\mathrm{tr}})$ training input points $\{\boldsymbol{x}_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}-\widetilde{n}_{\mathrm{tr}}}$ are generated independently following $p_{\mathrm{tr}}^{\mathrm{FBV}_{\mathrm{OW}}}(\boldsymbol{x})$ and corresponding training output values $\{y_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}-\widetilde{n}_{\mathrm{tr}}}$ are observed. Finally, the parameter $\boldsymbol{\theta}$ is learned using $\{(\widetilde{\boldsymbol{x}}_i^{\mathrm{tr}}, \widetilde{y}_i^{\mathrm{tr}})\}_{i=1}^{\widetilde{n}_{\mathrm{tr}}}$ and $\{(\boldsymbol{x}_i^{\mathrm{tr}}, y_i^{\mathrm{tr}})\}_{i=1}^{n_{\mathrm{tr}}-\widetilde{n}_{\mathrm{tr}}}$ as

$$\widehat{\boldsymbol{\theta}}_{\mathrm{OW}} = \underset{\boldsymbol{\theta}}{\mathrm{argmin}}\left[\sum_{i=1}^{\widetilde{n}_{\mathrm{tr}}}\left(\widehat{f}(\widetilde{\boldsymbol{x}}_i^{\mathrm{tr}}) - \widetilde{y}_i^{\mathrm{tr}}\right)^2 + \sum_{i=1}^{n_{\mathrm{tr}}-\widetilde{n}_{\mathrm{tr}}} w_{\mathrm{FBV}_{\mathrm{OW}}}(\boldsymbol{x}_i^{\mathrm{tr}})\left(\widehat{f}(\boldsymbol{x}_i^{\mathrm{tr}}) - y_i^{\mathrm{tr}}\right)^2\right], \tag{67}$$

where

$$w_{\mathrm{FBV}_{\mathrm{OW}}}(\boldsymbol{x}) = \frac{p_{\mathrm{te}}(\boldsymbol{x})}{p_{\mathrm{tr}}^{\mathrm{FBV}_{\mathrm{OW}}}(\boldsymbol{x})}. \tag{68}$$

The subscript ‘OW’ denotes ‘Ordinary and Weighted’. Note that $\mathrm{FBV}_{\mathrm{OW}}$ depends on the realization of $\{\widetilde{\boldsymbol{x}}_i^{\mathrm{tr}}\}_{i=1}^{\widetilde{n}_{\mathrm{tr}}}$, but is independent of the realization of $\{\boldsymbol{x}_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}-\widetilde{n}_{\mathrm{tr}}}$.

Kanamori and Shimodaira (2003) proved that for $\widetilde{n}_{\mathrm{tr}} = o(n_{\mathrm{tr}})$, $\lim_{n_{\mathrm{tr}}\to\infty}\widetilde{n}_{\mathrm{tr}} = \infty$, and $\delta = \mathcal{O}(1)$,

$$\underset{\{\boldsymbol{x}_i\}_{i=1}^{n_{\mathrm{tr}}}}{\mathbb{E}}\ \underset{\{\epsilon_i\}_{i=1}^{n_{\mathrm{tr}}}}{\mathbb{E}}\ G_{\mathrm{W}} = \mathrm{FBV}_{\mathrm{OW}} + o({n_{\mathrm{tr}}}^{-1}), \tag{69}$$

by which the use of $\mathrm{FBV}_{\mathrm{OW}}$ is justified. The order of δ required above is weaker than that required in ALICE or FV_{W}. Therefore, $\mathrm{FBV}_{\mathrm{OW}}$ has a wider range of applications than ALICE or FV_{W}. However, this property may not be practically so valuable since learning with totally misspecified models (i.e., $\delta = \mathcal{O}(1)$) may not work well due to large model error (e.g., when a highly non-linear function is approximated by a straight line). Furthermore, the fact that $\widetilde{n}_{\mathrm{tr}}$ training input points should be gathered following $p_{\mathrm{te}}(\boldsymbol{x})$ in the first stage implies that we are only allowed to optimize the location of $n_{\mathrm{tr}} - \widetilde{n}_{\mathrm{tr}}$ remaining training input points. This is highly restrictive when the total number n_{tr} is not so large, which would be a usual case in active learning (e.g., Coomans et al. 1983; Baldi and Brunak 1998; Warmuth et al. 2003).

It was shown that the optimal training input density $p_{\mathrm{tr}}^{\mathrm{FBV}_{\mathrm{OW}}}(\boldsymbol{x})$ can be expressed in a closed-form as follows (Kanamori and Shimodaira 2003; Kanamori 2007):

$$p_{\mathrm{tr}}^{\mathrm{FBV}_{\mathrm{OW}}}(\boldsymbol{x}) \propto p_{\mathrm{te}}(\boldsymbol{x}) b_{\mathrm{FBV}_{\mathrm{OW}}}(\boldsymbol{x}), \tag{70}$$

where

$$b_{\mathrm{FBV}_{\mathrm{OW}}}(\boldsymbol{x}) = \left(\sum_{\ell,\ell'=1}^{t}[\boldsymbol{U}^{-1}]_{\ell,\ell'}\varphi_{\ell}(\boldsymbol{x})\varphi_{\ell'}(\boldsymbol{x})(\delta^2 r^2(\boldsymbol{x}) + \sigma^2)\right)^{\frac{1}{2}}. \tag{71}$$

However, since $(\delta^2 r^2(\boldsymbol{x}) + \sigma^2)$ is inaccessible, the above closed-form cannot be directly used for active learning. To cope with this problem, Kanamori (2007) proposed using a regression method. It can be shown that a consistent estimate of the value of $\left(b_{\mathrm{FBV}_{\mathrm{OW}}}(\boldsymbol{x})\right)^2$ at $\widetilde{\boldsymbol{x}}_i^{\mathrm{tr}}$ $(i = 1, 2, \ldots, \widetilde{n}_{\mathrm{tr}})$ is given by $[\widetilde{\boldsymbol{Q}}^2\widetilde{\boldsymbol{X}}\widetilde{\boldsymbol{U}}^{-1}\widetilde{\boldsymbol{X}}^{\top}]_{i,i}$. Based on the input-output samples $\{(\widetilde{\boldsymbol{x}}_i^{\mathrm{tr}}, [\widetilde{\boldsymbol{Q}}^2\widetilde{\boldsymbol{X}}\widetilde{\boldsymbol{U}}^{-1}\widetilde{\boldsymbol{X}}^{\top}]_{i,i})\}_{i=1}^{\widetilde{n}_{\mathrm{tr}}}$, a regression method is used for *learning* the function $\left(b_{\mathrm{FBV}_{\mathrm{OW}}}(\boldsymbol{x})\right)^2$. Let us denote the learned function by $\widehat{b}_{\mathrm{FBV}_{\mathrm{OW}}}(\boldsymbol{x})$. Then the optimal training input density and the importance weight are approximated as

$$\widehat{p}_{\mathrm{tr}}^{\mathrm{FBV}_{\mathrm{OW}}}(\boldsymbol{x}) \propto p_{\mathrm{te}}(\boldsymbol{x})\widehat{b}_{\mathrm{FBV}_{\mathrm{OW}}}(\boldsymbol{x}), \tag{72}$$

$$\widehat{w}_{\mathrm{FBV}_{\mathrm{OW}}}(\boldsymbol{x}) \propto \frac{1}{\widehat{b}_{\mathrm{FBV}_{\mathrm{OW}}}(\boldsymbol{x})}. \tag{73}$$

Since the value of $\widehat{b}_{\mathrm{FBV}_{\mathrm{OW}}}(\boldsymbol{x})$ is available at any input location $\boldsymbol{x}$, $\{\widehat{b}_{\mathrm{FBV}_{\mathrm{OW}}}(\boldsymbol{x}_j^{\mathrm{te}})\}_{j=1}^{n_{\mathrm{te}}}$ can be computed and used as a resampling bias function in pool-based active learning. However, this method still suffers from the limitations caused by the two-stage approach pointed out above. Furthermore, obtaining a good approximation $\widehat{b}_{\mathrm{P\text{-}FBV}_{\mathrm{OW}}}(\boldsymbol{x})$ by regression is generally difficult; thus P-FBV$_{\mathrm{OW}}$ may not be so reliable in practice.

5 Simulations

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

5.1 Toy dataset

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

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

$$f(x) = 1 - x + x^2 + \delta r(x), \tag{74}$$

where

$$r(x) = \frac{z^3 - 3z}{\sqrt{6}} \quad \text{with } z = \frac{x - 0.2}{0.4}. \tag{75}$$

Note that the above $r(x)$ is the Hermite polynomial, which ensures the orthonormality of $r(x)$ to the 2nd order polynomial model under a Gaussian test input distribution (see below for detail). Let us consider the following three cases.

$$\delta = 0, 0.03, 0.06. \tag{76}$$

See the top graph of Fig. 5 for the profiles of $f(x)$ with different δ.

Let the number of training samples to gather be $n_{\mathrm{tr}} = 100$ and let $\{\epsilon_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ be i.i.d. Gaussian noise with mean zero and standard deviation $\sigma = 0.3$, where σ is treated as unknown here. Let the test input density $p_{\mathrm{te}}(x)$ be the Gaussian density with mean 0.2 and standard deviation 0.4; $p_{\mathrm{te}}(x)$ is also treated as unknown here. See the bottom graph of Fig. 5 for the profile of $p_{\mathrm{te}}(x)$. Let us draw $n_{\mathrm{te}} = 1000$ test input points independently from the test input distribution.

A polynomial model of order 2 is used for learning:

$$\widehat{f}(x) = \theta_1 + \theta_2 x + \theta_3 x^2. \tag{77}$$

Note that for these basis functions, the residual function $r(x)$ in (75) fulfills the orthogonality condition (19) and normalization condition (20).

In this experiment, we compare the performance of the following sampling strategies:

P-ALICE: Training input points are drawn following (14) for

$$\lambda \in \Lambda_{\mathrm{coarse}} \cup \Lambda_{\mathrm{fine}}, \tag{78}$$

where

$$\Lambda_{\mathrm{coarse}} = \{0, 0.1, 0.2 \ldots, 1\}, \tag{79}$$

$$\Lambda_{\mathrm{fine}} = \{0.4, 0.41, 0.42, \ldots, 0.6\}. \tag{80}$$

Then the best value of λ is chosen from the above candidates based on (11). IWLS is used for parameter learning.

P-FV$_{\mathrm{W}}$: Training input points are drawn following (44) (or equivalently (14) with $\lambda = 0.5$). IWLS is used for parameter learning.

P-CV$_{\mathrm{O}}$: Training input points are drawn following (14) for (78) and the best value of λ is chosen based on (56). OLS is used for parameter learning.

P-FBV$_{\mathrm{OW}}$: Initially, 50 training input-output samples are gathered based on the test input distribution and they are used for learning the resampling bias function $b_{\mathrm{FBV_{OW}}}(\boldsymbol{x})$; the resampling bias function is learned by kernel ridge regression with Gaussian kernels, where the Gaussian width and ridge parameter are optimized based on 5-fold cross-validation by exhaustive grid search. Then the remaining 50 training input points are chosen based on (72). OLS+IWLS is used for parameter learning (see (67)).

Passive: Training input points are drawn uniformly from the pool of test input samples (or equivalently (14) with $\lambda = 0$). OLS is used for parameter learning.

For references, the profile of $p_{\mathrm{tr}}^{\mathrm{FV_W}}(x)$ (the optimal training input density by FV$_{\mathrm{W}}$; see (40)) is also depicted in the bottom graph of Fig. 5.

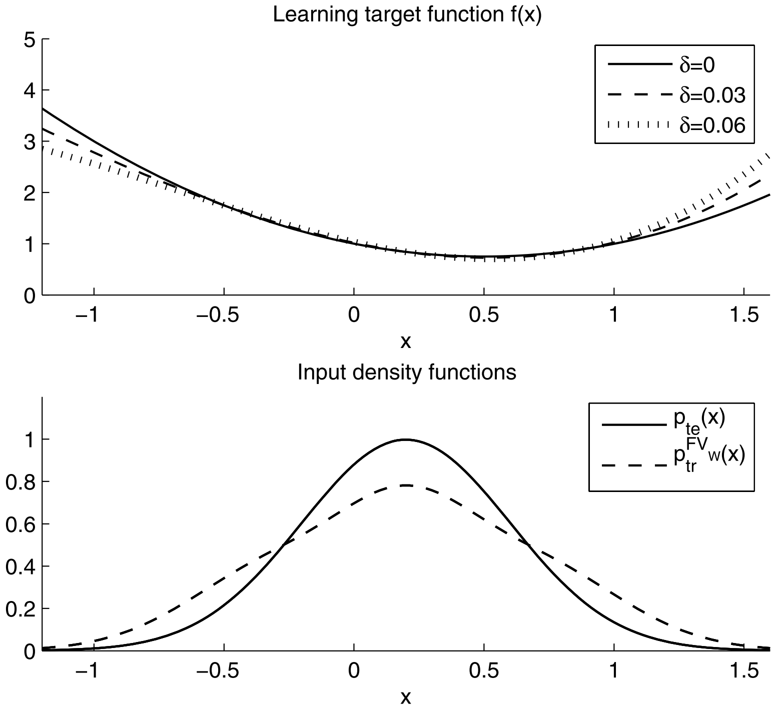


Fig. 5 Learning target function and test input density function

Table 1 The mean squared test error for the toy dataset (means and standard deviations over 100 trials). For better comparison, the model error δ^2 is subtracted from the error and all values are multiplied by 10^3. In each row of the table, the best method and comparable ones by the Wilcoxon signed rank test at the significance level 5% are indicated with '∘'

	P-ALICE	P-FV$_{\mathrm{W}}$	P-CV$_{\mathrm{O}}$	P-FBV$_{\mathrm{OW}}$	Passive
$\delta = 0$	$^\circ 2.03 \pm 1.81$	2.59 ± 1.83	$^\circ 1.82 \pm 1.69$	6.43 ± 6.61	3.10 ± 3.09
$\delta = 0.03$	$^\circ 2.17 \pm 2.04$	2.81 ± 2.01	2.62 ± 2.05	6.66 ± 6.54	3.40 ± 3.55
$\delta = 0.06$	$^\circ 2.42 \pm 2.65$	3.19 ± 2.59	4.85 ± 3.37	7.65 ± 7.21	4.12 ± 4.71
Average	$^\circ 2.21 \pm 2.19$	2.86 ± 2.18	3.10 ± 2.78	6.91 ± 6.79	3.54 ± 3.85

In Table 1, the mean squared test error (2) obtained by each method is described. The numbers in the table are means and standard deviations over 100 trials. For better comparison, the model error δ^2 is subtracted from the obtained error and all values are multiplied by 10^3. In each row of the table, the best method and comparable ones by the *Wilcoxon signed rank test* (e.g., Henkel 1979) at the significance level 5% are indicated with '∘'.

When $\delta = 0$, P-CV$_{\mathrm{O}}$ works the best and is followed by P-ALICE. These two methods have no statistically significant difference and are significantly better than the other methods. When δ is increased from 0 to 0.03, the performance of P-ALICE and P-FV$_{\mathrm{W}}$ is almost unchanged, while the performance of P-CV$_{\mathrm{O}}$ is considerably degraded. Consequently, P-ALICE gives the best performance among all. When δ is further increased to 0.06, the performance of P-ALICE and P-FV$_{\mathrm{W}}$ are still almost unchanged. On the other hand, P-CV$_{\mathrm{O}}$ performs very poorly and is outperformed even by the baseline Passive method. P-FBV$_{\mathrm{OW}}$ does not seem to work well for all three cases.

Overall, P-ALICE and P-FV$_{\mathrm{W}}$ are shown to be highly robust against model misspecification, while P-CV$_{\mathrm{O}}$ is very sensitive to the violation of the model correctness assumption. P-ALICE significantly outperforms P-FV$_{\mathrm{W}}$, which would be caused by the fact that ALICE is a more accurate estimator of the single-trial generalization error than FV$_{\mathrm{W}}$ (see Sect. 3.6).

5.2 Benchmark datasets

The *Bank*, *Kin*, and *Pumadyn* regression benchmark data families provided by DELVE (Rasmussen et al. 1996) are used here. Each data family consists of 8 different datasets:

Input dimension d: Input dimension is either $d = 8$ or 32.
Target function type: The target function is either 'fairly linear' or 'non-linear' ('f' or 'n').
Unpredictability/noise level: The unpredictability/noise level is either 'medium' or 'high' ('m' or 'h').

Thus 24 datasets are used in total. Each dataset includes 8192 samples, consisting of d-dimensional input and 1-dimensional output data. For convenience, every attribute is normalized into [0, 1].

All 8192 input samples are used as the pool of test input points (i.e., $n_{\mathrm{te}} = 8192$) and $n_{\mathrm{tr}} = 100$ training input points are chosen from the pool when $d = 8$; $n_{\mathrm{tr}} = 300$ training input points are chosen when $d = 32$. The following linear regression model is used for learning:

$$\widehat{f}(\boldsymbol{x}) = \sum_{\ell=1}^{50} \theta_\ell \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{c}_\ell\|^2}{2}\right), \tag{81}$$

Table 2 The mean squared test error (2) for 8-dimensional benchmark datasets (means and standard deviations over 1000 trials). For better comparison, all the values are normalized by the mean error of the Passive method. The best method and comparable ones by the Wilcoxon signed rank test at the significance level 5% are indicated with '∘'

	P-ALICE	P-FV$_\mathrm{W}$	P-CV$_\mathrm{O}$	ALICE	Passive
bank-8fm	∘0.89 ± 0.14	0.95 ± 0.16	0.91 ± 0.14	1.16 ± 0.26	1.00 ± 0.19
bank-8fh	0.86 ± 0.14	0.94 ± 0.17	∘0.85 ± 0.14	0.97 ± 0.20	1.00 ± 0.20
bank-8nm	∘0.89 ± 0.16	0.95 ± 0.20	0.91 ± 0.18	1.18 ± 0.28	1.00 ± 0.21
bank-8nh	0.88 ± 0.16	0.95 ± 0.20	∘0.87 ± 0.16	1.02 ± 0.28	1.00 ± 0.21
kin-8fm	0.78 ± 0.22	0.87 ± 0.24	0.87 ± 0.22	∘0.39 ± 0.20	1.00 ± 0.25
kin-8fh	0.80 ± 0.17	0.88 ± 0.21	0.85 ± 0.17	∘0.54 ± 0.16	1.00 ± 0.23
kin-8nm	∘0.91 ± 0.14	0.97 ± 0.16	0.92 ± 0.14	0.97 ± 0.18	1.00 ± 0.17
kin-8nh	∘0.90 ± 0.13	0.96 ± 0.16	0.90 ± 0.13	0.95 ± 0.17	1.00 ± 0.17
pumadyn-8fm	∘0.89 ± 0.13	0.95 ± 0.16	∘0.89 ± 0.12	0.93 ± 0.16	1.00 ± 0.18
pumadyn-8fh	0.89 ± 0.13	0.98 ± 0.16	∘0.88 ± 0.12	0.93 ± 0.15	1.00 ± 0.17
pumadyn-8nm	∘0.91 ± 0.13	0.98 ± 0.17	0.92 ± 0.13	1.03 ± 0.18	1.00 ± 0.18
pumadyn-8nh	∘0.91 ± 0.13	0.97 ± 0.14	0.91 ± 0.13	0.98 ± 0.16	1.00 ± 0.17
Average	∘0.87 ± 0.16	0.95 ± 0.18	0.89 ± 0.15	0.92 ± 0.30	1.00 ± 0.20

where $\{\boldsymbol{c}_\ell\}_{\ell=1}^{50}$ are template points randomly chosen from the pool of test input points. Other settings are the same as the toy experiments in Sect. 5.1.

In addition to the pool-based methods P-ALICE, P-FV$_\mathrm{W}$, and P-CV$_\mathrm{O}$, the population-based method ALICE is also tested here. In this experiment, the test input density $p_{\mathrm{te}}(\boldsymbol{x})$ is unknown. So it is estimated using the uncorrelated multi-dimensional Gaussian density:

$$p_{\mathrm{te}}(\boldsymbol{x}) = \frac{1}{(2\pi\widehat{\gamma}_{\mathrm{MLE}}^2)^{\frac{d}{2}}} \exp\left(-\frac{\|\boldsymbol{x}-\widehat{\boldsymbol{\mu}}_{\mathrm{MLE}}\|^2}{2\widehat{\gamma}_{\mathrm{MLE}}^2}\right), \tag{82}$$

where $\widehat{\boldsymbol{\mu}}_{\mathrm{MLE}}$ and $\widehat{\gamma}_{\mathrm{MLE}}$ are the maximum likelihood estimates of the mean and standard deviation obtained from all 8192 unlabeled samples. The training input density $p_{\mathrm{tr}}(\boldsymbol{x})$ from the set of uncorrelated multi-dimensional Gaussian densities with mean $\widehat{\boldsymbol{\mu}}_{\mathrm{MLE}}$ and standard deviation $c\widehat{\gamma}_{\mathrm{MLE}}$, where

$$c = 0.7, 0.8, 0.9, \ldots, 2.4. \tag{83}$$

Based on the training input density determined by a population-based method, input points are chosen from the pool of unlabeled samples as follows. First, provisional input points are created following the chosen training input density. Then the input points in the pool of unlabeled samples that are closest to the provisional input points are chosen without overlap.

Tables 2 and 3 summarize the mean squared test error (2) for $d = 8$ and 32, respectively. The numbers are the means and standard deviations over 1000 trials. For better comparison, all the values are normalized by the mean error of the Passive method. The best method and comparable ones by the Wilcoxon signed rank test at the significance level 5% are indicated with '∘'.

When $d = 8$, all 3 pool-based active learning methods outperform the Passive method. Among them, P-ALICE tends to significantly outperform P-FV$_\mathrm{W}$ and P-CV$_\mathrm{O}$. The population-based method ALICE works rather well, but it is not as good as the pool-based

Table 3 The mean squared test error (2) for 32-dimensional benchmark datasets (means and standard deviations over 1000 trials)

	P-ALICE	P-FV$_\mathrm{W}$	P-CV$_\mathrm{O}$	ALICE	Passive
bank-32fm	0.97 ± 0.05	0.99 ± 0.05	°0.96 ± 0.04	1.04 ± 0.06	1.00 ± 0.06
bank-32fh	0.98 ± 0.05	0.99 ± 0.05	°0.96 ± 0.04	1.01 ± 0.05	1.00 ± 0.05
bank-32nm	0.98 ± 0.06	0.99 ± 0.07	°0.96 ± 0.06	1.03 ± 0.07	1.00 ± 0.07
bank-32nh	0.97 ± 0.05	0.99 ± 0.06	°0.96 ± 0.05	0.99 ± 0.05	1.00 ± 0.06
kin-32fm	°0.79 ± 0.07	0.93 ± 0.09	1.53 ± 0.14	0.98 ± 0.09	1.00 ± 0.11
kin-32fh	°0.79 ± 0.07	0.92 ± 0.08	1.40 ± 0.12	0.98 ± 0.09	1.00 ± 0.10
kin-32nm	0.95 ± 0.04	0.97 ± 0.04	°0.93 ± 0.04	1.03 ± 0.05	1.00 ± 0.05
kin-32nh	0.95 ± 0.04	0.97 ± 0.04	°0.92 ± 0.03	1.02 ± 0.04	1.00 ± 0.05
pumadyn-32fm	0.98 ± 0.12	0.99 ± 0.13	1.15 ± 0.15	°0.96 ± 0.12	1.00 ± 0.13
pumadyn-32fh	0.96 ± 0.04	0.98 ± 0.05	°0.95 ± 0.04	0.97 ± 0.04	1.00 ± 0.05
pumadyn-32nm	0.96 ± 0.04	0.98 ± 0.04	°0.93 ± 0.03	0.96 ± 0.03	1.00 ± 0.05
pumadyn-32nh	0.96 ± 0.03	0.98 ± 0.04	°0.92 ± 0.03	0.97 ± 0.04	1.00 ± 0.04
Average	°0.94 ± 0.09	0.97 ± 0.07	1.05 ± 0.21	1.00 ± 0.07	1.00 ± 0.07

counterpart P-ALICE. This would be the fruit of directly defining the training distribution over unlabeled samples.

When $d = 32$, P-CV$_\mathrm{O}$ outperforms P-ALICE and P-FV$_\mathrm{W}$ for many datasets. However, the performance of P-CV$_\mathrm{O}$ is unstable and it works very poorly for the *kin32-fm*, *kin32-fh*, and *pumadyn32-fm* datasets. Consequently, the average error of P-CV$_\mathrm{O}$ over all 12 datasets is worse than the baseline Passive sampling scheme. On the other hand, P-ALICE and P-FV$_\mathrm{W}$ are still stable and consistently outperform the Passive method. Among these two methods, P-ALICE tends to outperform P-FV$_\mathrm{W}$. The population-based method ALICE tends to be outperformed by the pool-based counterpart P-ALICE.

P-ALICE and P-FV$_\mathrm{W}$ are shown to be more reliable than P-CV$_\mathrm{O}$, and P-ALICE tends to outperform P-FV$_\mathrm{W}$. When the input dimension is high, the variance tends to dominate the bias due to sparsity of data points. Then the bias caused by model misspecification is no longer critical and therefore P-CV$_\mathrm{O}$ tends to be better. However, P-CV$_\mathrm{O}$ has catastrophic cases—which would be the situation where the bias is not negligibly small even in high-dimensional cases. This is consistent with the illustrative experiments shown in Sect. 5.1.

Overall, the proposed method P-ALICE is shown to be robust against such catastrophic cases even in high-dimensional cases and therefore would be more reliable in practice.

6 Real-world applications

Finally, we apply the proposed active learning method to a wafer alignment problem in semiconductor exposure apparatus (see Fig. 6).

Recent semiconductors have the layered circuit structure, which are built by exposing circuit patterns multiple times. In this process, it is extremely important to align the wafer at the same position with very high accuracy. To this end, the location of markers are measured to adjust the shift and rotation of wafers. However, measuring the location of markers is time-consuming and therefore there is a strong need to reduce the number of markers to measure for speeding up the semiconductor production process.

Fig. 6 Semiconductor exposure apparatus

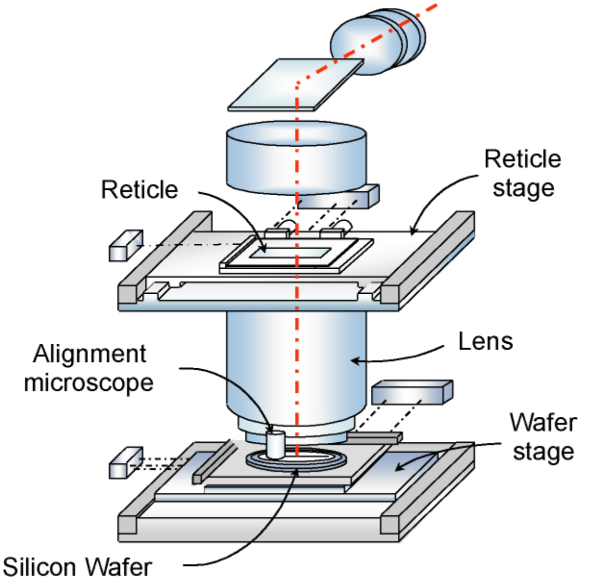


Fig. 7 Silicon wafer with markers. Observed markers based on the conventional heuristic are also shown

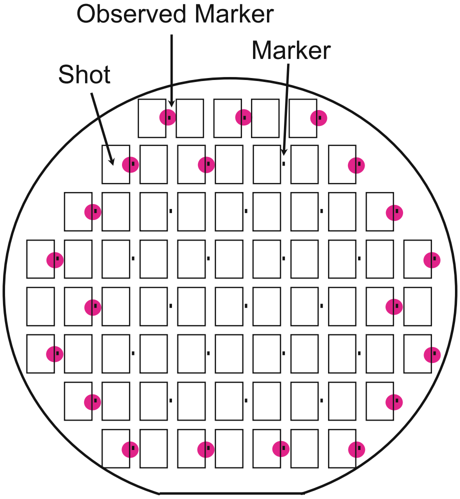


Figure 7 illustrates a wafer, where markers are printed uniformly over the wafer. Our goal here is to choose the most 'informative' markers to measure for better alignment of the wafer. A conventional choice is to measure markers far from the center in a symmetric way, which would provide robust estimation of the rotation angle (see Fig. 7). However, this naive approach is not necessarily the best since misalignment is not only caused by affine transformation, but also by several other non-linear factors such as a warp, a biased characteristic of measurement apparatus, and different temperature conditions. In practice, it is not easy to model such non-linear factors accurately, so the linear affine model or the second-order model is often used in wafer alignment. However, this causes model misspecification and therefore our proposed active learning method would be useful in this application.

Table 4 The mean squared test error for the wafer alignment problem (means and standard deviations over 220 wafers). 'Conv.' indicates the conventional heuristic of choosing the outer markers

Order	P-ALICE	P-FV$_W$	P-CV$_O$	Passive	Conv.
1	°2.27 ± 1.08	2.29 ± 1.08	2.37 ± 1.15	2.32 ± 1.11	2.36 ± 1.15
2	°1.93 ± 0.89	2.09 ± 0.98	1.96 ± 0.91	2.32 ± 1.15	2.13 ± 1.08

Let us consider the functions whose input $\boldsymbol{x} = (u, v)^\top$ is the location on the wafer and whose output is the horizontal discrepancy Δu or the vertical discrepancy Δv. These functions are learned using the following second-order model.

$$\Delta u \text{ or } \Delta v = \theta_0 + \theta_1 u + \theta_2 v + \theta_3 uv + \theta_4 u^2 + \theta_5 v^2. \tag{84}$$

We totally have 220 wafer samples and our experiment is carried out as follows. For each wafer, $n_{\mathrm{tr}} = 20$ points are chosen from $n_{\mathrm{te}} = 38$ markers and the horizontal and the vertical discrepancies are observed. Then the above model is trained and its prediction performance is tested using all 38 markers in the 220 wafers. This process is repeated for all 220 wafers. Since the choice of the sampling location by active learning methods is stochastic, the above experiment is repeated for 100 times with different random seeds.

The mean and standard deviation of the squared test error over 220 wafers are summarized in Table 4. This shows that the proposed P-ALICE method works significantly better than the other sampling strategies and it provides about 10-percent reduction in the squared error from the conventional heuristic of choosing the outer markers. Similar experiments have also been conducted using the first-order model and confirmed that P-ALICE still works the best.

7 Conclusions and outlook

We extended a population-based active learning method (FV$_W$) to a pool-based scenario (P-FV$_W$) and derived a closed-form 'optimal' resampling bias function. This closed-form solution is optimal within the full-expectation framework, but is not necessarily optimal in the single-trial analysis. To further improve the performance, we extended another population-based method (ALICE) to a pool-based scenario (P-ALICE). Since ALICE is derived within the conditional-expectation framework and therefore input-dependent, it is provably more accurate than FV$_W$ in the single-trial analysis. However, P-ALICE does not allow us to obtain a closed-form solution due to its input-dependence. To cope with this problem, we proposed a practical heuristic procedure which efficiently searches for a better solution around the P-FV$_W$ optimal solution. Numerical experiments with toy and benchmark datasets showed that the proposed method consistently outperforms the baseline passive learning scheme and compares favorably with other active learning methods. Furthermore, the usefulness of the proposed active learning method was also demonstrated in wafer alignment in semiconductor exposure apparatus.

In P-ALICE, a reasonable candidate set of resampling bias functions needs to be prepared. In this paper, (14) was chosen as a heuristic and was shown to be reasonable through experiments. Even so, there is still room for further improvement and it is important to find alternative strategies for preparing better candidates.

We focused on regression scenarios in this paper. A natural desire is to extend the same idea to classification scenarios. We expect that the conceptual issues we addressed in this paper—the usefulness of the conditional-expectation approach and the practical importance of dealing with approximate correct models (Sect. 3)—are still valid in classification scenarios. In the future, we will explore active learning problems in classification scenarios based on these conceptual ideas.

The P-ALICE criterion is a random variable which depends not only on training input distributions, but also on realizations of training input points. This is why the minimizer of P-ALICE cannot be obtained analytically; we resorted to a greedy search around the solution of P-FV$_{\mathrm{W}}$. On the other hand, this fact implies that the P-ALICE criterion allows us to evaluate the goodness of not only training input distributions but also realizations of training input points. We conducted preliminary experiments in which training input points are drawn several times from the same training input distribution and experienced that the experimental performance is sometimes further improved by multiple draws. Thus it would be interesting to investigate this phenomenon more systematically. This issue seems to be related to the sequential design of experiments and therefore further study along this line would be fruitful.

Our active learning method is valid for approximately correct models, which is an advantage over traditional OLS-based active learning methods. However, when the model is totally misspecified, it is necessary to perform model selection (e.g., Shimodaira 2000; Sugiyama and Müller 2005; Sugiyama et al. 2007) since large model error will dominate the bias and variance, and therefore learning with such a totally misspecified model is not useful in practice. However, performing model selection and active learning at the same time, which is called *active learning with model selection*, is not straightforward due to the *active learning/model selection dilemma* (Sugiyama and Ogawa 2003).

- In order to select training input points by an existing active learning method, a model must have been fixed (i.e., model selection must have been performed).
- In order to select the model by a standard model selection method, the training input points must have been fixed (i.e., active learning must have been performed).

To cope with this dilemma, a novel approach has been explored recently (Sugiyama and Rubens 2008). However, the existing study focuses on population-based scenarios and active learning with model selection under pool-based settings seems to still be an open research issue. We expect that the result given in this paper could be a basis for further investigating this challenging topic.

The proposed method has been shown to be robust against the existence of bias. However, if the input dimensionality is very high, the variance tends to dominate the bias due to sparsity of data samples and therefore the advantage of the proposed method tends to be lost. Moreover critically, regression from data samples is highly unreliable in such high-dimensional problems due to extremely large variance. To address this issue, it would be important to first reduce the dimensionality of the data, which is another challenge in active learning research. For classification active learning in high dimensional problems, see Melville and Mooney (2004) and Schein and Ungar (2007).

We have focused on linear models. However, the importance weighting technique used for compensating for the bias caused by model misspecification is valid for any empirical-error based methods (Sugiyama et al. 2007). Thus another important direction to be pursued would be to extend the current active learning idea to more complex models such as support vector machines (Vapnik 1998) and neural networks (Bishop 1995).

Acknowledgements The author would like to thank Steffen Bickel and Tobias Scheffer for their helpful comments on our earlier manuscript. Our gratitude also goes to anonymous reviewers for their fruitful comments. This work was supported by MEXT (17700142 and 18300057), the Microsoft CORE3 Project, and the IBM Faculty Award.

References

Bach, F. R. (2007). Active learning for misspecified generalized linear models. In B. Schölkopf, J. Platt, & T. Hoffman (Eds.), *Advances in neural information processing systems* (Vol. 19, pp. 65–72). Cambridge: MIT Press.

Baldi, P., & Brunak, S. (1998). *Bioinformatics: the machine learning approach*. Cambridge: MIT Press.

Bickel, S., Brückner, M., & Scheffer, T. (2007). Discriminative learning for differing training and test distributions. In *Proceedings of the 24th international conference on machine learning*.

Bishop, C. M. (1995). *Neural networks for pattern recognition*. Oxford: Clarendon.

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

Coomans, D., Broeckaert, M., Jonckheer, M., & Massart, D. L. (1983). Comparison of multivariate discriminant techniques for clinical data—application to the thyroid functional state. *Methods of Information in Medicine*, *22*, 93–101.

Fedorov, V. V. (1972). *Theory of optimal experiments*. San Diego: Academic Press.

Fishman, G. S. (1996). *Monte Carlo: concepts, algorithms, and applications*. Berlin: Springer.

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

Henkel, R. E. (1979). *Tests of significance*. Thousand Oaks: SAGE.

Huang, J., Smola, A., Gretton, A., Borgwardt, K. M., & Schölkopf, B. (2007). Correcting sample selection bias by unlabeled data. In B. Schölkopf & J. Platt (Eds.), *Advances in neural information processing systems* (Vol. 19, pp. 601–608). Cambridge: MIT Press.

Kanamori, T. (2007). Pool-based active learning with optimal sampling distribution and its information geometrical interpretation. *Neurocomputing*, *71*(1–3), 353–362.

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

McCallum, A., & Nigam, K. (1998). Employing EM in pool-based active learning for text classification. In *Proceedings of the 15th international conference on machine learning*.

Melville, P., & Mooney, R. J. (2004). Diverse ensembles for active learning. In *Proceedings of the twenty-first international conference on machine learning*. New York: ACM.

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

Schein, A. I., & Ungar, L. H. (2007). Active learning for logistic regression: an evaluation. *Machine Learning*, *68*(3), 235–265.

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

Sugiyama, M. (2006). Active learning in approximately linear regression based on conditional expectation of generalization error. *Journal of Machine Learning Research*, *7*, 141–166.

Sugiyama, M., & Müller, K.-R. (2005). Input-dependent estimation of generalization error under covariate shift. *Statistics & Decisions*, *23*(4), 249–279.

Sugiyama, M., & Ogawa, H. (2003). 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.

Sugiyama, M., & Rubens, N. (2008). A batch ensemble approach to active learning with model selection. *Neural Networks*, *21*(9), 1287–1286.

Sugiyama, M., Krauledat, M., & Müller, K.-R. (2007). Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, *8*, 985–1005.

Sugiyama, M., Suzuki, T., Nakajima, S., Kashima, H., von Bünau, P., & Kawanabe, M. (2008). Direct importance estimation for covariate shift adaptation. *Annals of the Institute of Statistical Mathematics*, *60*(4), 699–746.

Sugiyama, M., Rubens, N., & Müller, K.-R. (2009). A conditional expectation approach to model selection and active learning under covariate shift. In J. Quiñonero-Candela, M. Sugiyama, A. Schwaighofer, & N. Lawrence (Eds.), *Dataset shift in machine learning* (pp. 107–130). Cambridge: MIT Press.

Vapnik, V. N. (1998). *Statistical learning theory*. New York: Wiley.

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

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