Improving Importance Estimation in Pool-based Batch Active Learning for Approximate Linear Regression

Nozomi Kurihara and Masashi Sugiyama
Department of Computer Science, Tokyo Institute of Technology
2-12-1-W8-74 O-okayama, Meguro-ku, Tokyo 152-8552, Japan
sugi@cs.titech.ac.jp

Abstract

Pool-based batch active learning is aimed at choosing training inputs from a 'pool' of test inputs so that the generalization error is minimized. P-ALICE (Pool-based Active Learning using Importance-weighted least-squares learning based on Conditional Expectation of the generalization error) is a state-of-the-art method that can cope with model misspecification by weighting training samples according to the importance (i.e., the ratio of test and training input densities). However, importance estimation in the original P-ALICE is based on the assumption that the number of training samples to gather is small, which is not always true in practice. In this paper, we propose an alternative scheme for importance estimation based on the inclusion probability, and show its validity through numerical experiments.

Keywords

pool-based batch active learning, approximate linear regression, covariate shift, importance-weighted least-squares, P-ALICE, inclusion probability

1 Introduction

The objective of supervised learning is to find an input-output relationship behind training samples (Hastie et al., 2001; Bishop, 2006). Once the input-output relationship is successfully learned, outputs for unseen inputs can be predicted, i.e., the learning machine can *generalize*.

When users are allowed to choose the location of training inputs, it is desirable to design the input locations so that the generalization error is minimized. Such a problem is called *active learning* (Settles, 2009) or *experiment design* (Fedorov, 1972; Pukelsheim, 1993), and has been shown to be useful in various application areas such as text classification (Lewis & Gale, 1994; McCallum & Nigam, 1998), age estimation from images (Ueki et al., 2010), medical data analysis (Wiens & Guttag, 2010), chemical data analysis (Warmuth et al., 2003), biological data analysis (Liu, 2004), and robot control (Akiyama et al., 2010).

If users are allowed to locate training inputs at any position in the domain, the active learning setup is said to be *population-based* (Wiens, 2000; Kanamori & Shimodaira, 2003; Sugiyama, 2006). On the other hand, if users need to choose training input locations from a pool of finite candidate points, it is said to be *pool-based* (McCallum & Nigam, 1998; Kanamori, 2007; Sugiyama & Nakajima, 2009). Depending on the way training input locations are chosen, active learning is also categorized into *sequential* or *batch* approaches: Training inputs are selected one by one iteratively in the sequential approach (Box & Hunter, 1965; Sugiyama & Ogawa, 2000), while all training inputs are selected at once in the batch approach (Kiefer, 1959; Sugiyama & Ogawa, 2001). In this paper, we focus on a pool-based batch active learning.

Active learning generally induces a *covariate shift*—a situation where training and test input distributions are different (Shimodaira, 2000; Quiñonero-Candela et al., 2009; Sugiyama & Kawanabe, 2012). When a model is correctly specified, covariate shifts do not matter in designing active learning methods. However, for a misspecified model, a covariate shift causes a strong estimation bias and thus classical active learning techniques that require a correct model become unreliable (Kiefer, 1959; Fedorov, 1972).

To cope with the bias induced by the covariate shift, active learning techniques that explicitly take model misspecification into account have been developed (Wiens, 2000; Kanamori & Shimodaira, 2003; Sugiyama, 2006; Kanamori, 2007; Sugiyama & Nakajima, 2009; Beygelzimer et al., 2009). The key idea of covariate shift adaptation is *importance weighting*—a loss function used for training is weighted according to the importance (i.e., the ratio of test and training input densities). Among the importance-weighted active learning methods, the pool-based batch active learning method for approximate linear regression called P-ALICE (Pool-based Active Learning using Importance-weighted least-squares learning based on Conditional Expectation of the generalization error) was demonstrated to be useful (Sugiyama & Nakajima, 2009).

However, in the original P-ALICE, the number of training samples to gather is assumed to be sufficiently smaller than the size of the sample pool. However, when this assumption is not satisfied, the importance weight used in P-ALICE is not reliable. In this paper, we propose a new method to set the importance weight that does not rely on this assumption. Our new weighting scheme is based on the *inclusion probability* (Horvitz & Thompson, 1952), which allows us to precisely capture the relation between the training and test input distributions. Through experiments, we show that the active learning performance of P-ALICE can be improved by the proposed weighting method when the training sample size is relatively large.

The rest of this paper is structured as follows. In Section 2, we formulate the problem of pool-based active learning and give an overview of P-ALICE. In Section 3, we point out a limitation of importance estimation in P-ALICE, and propose an alternative method. In Section 4, experimental results on toy and benchmark datasets are reported. Finally, concluding remarks are given in Section 5.

2 Problem Formulation

In this section, we formulate the problem of pool-based active learning and briefly review the P-ALICE method.

2.1 Pool-Based Active Learning for Linear Regression

Let us consider a regression problem of learning a real-valued function $f(\boldsymbol{x})$ defined on $\mathcal{D} \subset \mathbb{R}^d$. For training input-output samples

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

where $\{\epsilon_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ are i.i.d. noise with mean zero and unknown variance σ^2, let us use the following linear regression model:

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

where $\{\varphi_\ell(\boldsymbol{x})\}_{\ell=1}^{t}$ are fixed linearly independent basis functions and $\{\theta_\ell\}_{\ell=1}^{t}$ are parameters to be learned. Let us denote the vector of parameters by $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_t)^\top$, where $^\top$ denotes the transpose.

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

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

where the subscript 'W' denotes 'Weighted'. Let $\boldsymbol{X}$ be the $n_{\mathrm{tr}} \times t$ matrix with the (i, ℓ)-th element

$$X_{i,\ell} = \varphi_\ell(\boldsymbol{x}_i^{\mathrm{tr}}), \tag{3}$$

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

$$W_{i,i} = w(\boldsymbol{x}_i^{\mathrm{tr}}). \tag{4}$$

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

$$\widehat{\boldsymbol{\theta}}_{\mathrm{W}} = \boldsymbol{L}_{\mathrm{W}} \boldsymbol{y}^{\mathrm{tr}}, \tag{5}$$

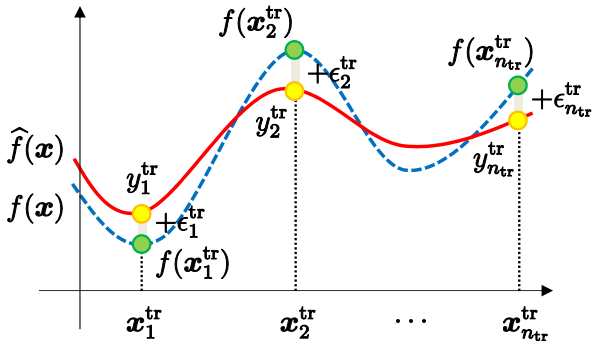


Figure 1: Regression problem.

where[1]

$$\boldsymbol{L}_{\mathrm{W}} = (\boldsymbol{X}^\top \boldsymbol{W} \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{W}, \tag{7}$$

$$\boldsymbol{y}^{\mathrm{tr}} = (y_1^{\mathrm{tr}}, y_2^{\mathrm{tr}}, \ldots, y_{n_{\mathrm{tr}}}^{\mathrm{tr}})^\top. \tag{8}$$

Note that the solution $\widehat{\boldsymbol{\theta}}_{\mathrm{W}}$ does not depend on the constant scaling of the weight function $w(\boldsymbol{x})$.

We adopt the squared-loss as the generalization error, i.e., the goodness of a learned function $\widehat{f}(\boldsymbol{x})$ is measured by

$$G = \int \left(\widehat{f}(\boldsymbol{x}^{\mathrm{te}}) - f(\boldsymbol{x}^{\mathrm{te}})\right)^2 p_{\mathrm{te}}(\boldsymbol{x}^{\mathrm{te}}) \mathrm{d}\boldsymbol{x}^{\mathrm{te}}, \tag{9}$$

where $p_{\mathrm{te}}(\boldsymbol{x})$ (> 0 for all $\boldsymbol{x} \in \mathcal{D}$) is a probability density function of test input points. The above formulation is summarized in Figure 1.

Below, we consider a pool-based active learning situation where we are given a 'pool' of test input points $\{\boldsymbol{x}_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$ drawn independently from $p_{\mathrm{te}}(\boldsymbol{x})$ and choose training input points $\{\boldsymbol{x}_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ from the pool so that the generalization error (9) is minimized.

2.2 P-ALICE

P-ALICE (Pool-based Active Learning using Importance-weighted least-squares learning based on Conditional Expectation of the generalization error; Sugiyama & Nakajima, 2009) is a pool-based active learning method that chooses training input points one by

[1]Although we can obtain the analytic solution for Eq.(2), we often face with numerical instability when computing the inverse of the matrix $\boldsymbol{X}^\top \boldsymbol{W} \boldsymbol{X}$ in Eq.(7). To avoid this problem, we practically employ a regularization technique (Hoerl & Kennard, 1970; Tikhonov & Arsenin, 1977; Poggio & Girosi, 1990), i.e., Eq.(7) is replaced with

$$\boldsymbol{L}_{\mathrm{W}} = (\boldsymbol{X}^\top \boldsymbol{W} \boldsymbol{X} + \gamma \boldsymbol{I}_t)^{-1} \boldsymbol{X}^\top \boldsymbol{W}, \tag{6}$$

where γ is a small positive scalar called the regularization parameter and $\boldsymbol{I}_t$ is the $t \times t$ identity matrix. In our experiments, we set $\gamma = 10^{-10}$.

one (i.e., sampling *without* replacement), with probability proportional to a user-designed *resampling bias function* $b(\boldsymbol{x})$. Mathematically, the resampling bias function is a strictly positive function defined over the pool of test input samples. P-ALICE finds the resampling bias function that minimizes a generalization error estimator

$$J = \mathrm{tr}(\widehat{\boldsymbol{U}}\boldsymbol{L}_{\mathrm{W}}\boldsymbol{L}_{\mathrm{W}}^{\top}),$$

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

$$\widehat{U}_{\ell,\ell'} = \frac{1}{n_{\mathrm{te}}} \sum_{j=1}^{n_{\mathrm{te}}} \varphi_{\ell}(\boldsymbol{x}_j^{\mathrm{te}})\varphi_{\ell'}(\boldsymbol{x}_j^{\mathrm{te}}), \tag{10}$$

and

$$w(\boldsymbol{x}_j^{\mathrm{te}}) \propto \frac{1}{b(\boldsymbol{x}_j^{\mathrm{te}})} \tag{11}$$

is used as a weight in WLS (2).

A more detailed description of P-ALICE is given in Appendix.

3 Improving Importance Estimation in P-ALICE

In this section, we point out a weakness of P-ALICE and propose an alternative approach.

3.1 Weakness of P-ALICE

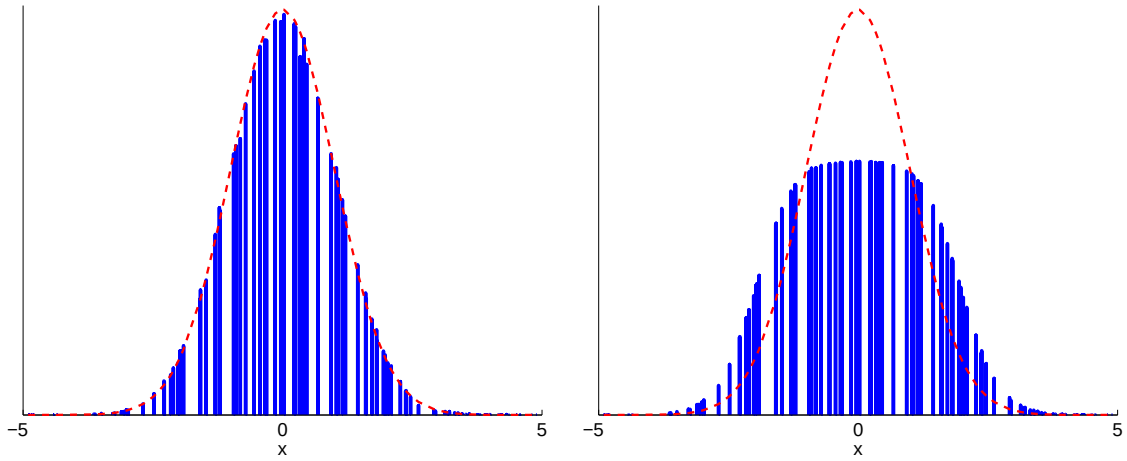
In P-ALICE, the importance weight $w(\boldsymbol{x}_j^{\mathrm{te}})$ is set as Eq.(11), which implies that the training input density $p_{\mathrm{tr}}(\boldsymbol{x}_j^{\mathrm{te}})$ is proportional to the product of the test input density $p_{\mathrm{te}}(\boldsymbol{x}_j^{\mathrm{te}})$ and a resampling bias function $b(\boldsymbol{x})$, i.e.,

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

Theoretically, the above derivation is based on the assumption that training input samples are i.i.d. However, because samples drawn *without* replacement from the pool are not generally i.i.d. and thus the above importance estimation method can be unreliable. In particular, when the size of the training set n_{tr} is not small relative to the pool size n_{te}, the influence of sampling without replacement is not negligible.

Let us illustrate this fact using a simple numerical example. Let 1-dimensional pool samples $\{x_j\}_{j=1}^{n_{\mathrm{te}}}$ be drawn from the uniform distribution on $[-5, 5]$ for $n_{\mathrm{te}} = 100$. The resampling bias function $b(x_j)$ is set to be proportional to the standard normal distribution:

$$b(x_j) \propto \exp\left(-\frac{x_j^2}{2}\right).$$



(a) $n_{\mathrm{tr}}/n_{\mathrm{te}} = 5/100$ (b) $n_{\mathrm{tr}}/n_{\mathrm{te}} = 40/100$

Figure 2: Illustration of the effect of sampling without replacement. 1-dimensional pool samples $\{x_j\}_{j=1}^{n_{\mathrm{te}}}$ are drawn from the uniform distribution on $[-5, 5]$ for $n_{\mathrm{te}} = 100$. The resampling bias function $b(x_j)$ is set to be proportional to the standard normal distribution. We draw $n_{\mathrm{tr}} = 5$ or $n_{\mathrm{tr}} = 40$ samples from the pool following $b(x_j)$ without replacement. In the graphs, the resampling bias function $b(x_j)$ is depicted by the dotted line and the frequency that the sample x_j was drawn over 10000 trials is depicted as a histogram.

We draw $n_{\mathrm{tr}} = 5$ or $n_{\mathrm{tr}} = 40$ samples from the pool following $b(x_j)$ without replacement. In Figure 2, $b(x_j)$ is depicted by the dotted line and the frequency that the sample x_j was drawn over 10000 trials is depicted as a histogram. We can see that, when n_{tr} is sufficiently smaller than n_{te} (i.e., $n_{\mathrm{tr}}/n_{\mathrm{te}} = 5/100$), the resampling bias function $b(x_j)$ well agrees with the histogram (i.e., the true distribution of drawn samples). On the other hand, when n_{tr} is not sufficiently smaller than n_{te} (i.e., $n_{\mathrm{tr}}/n_{\mathrm{te}} = 40/100$), the resampling bias function $b(x_j)$ is significantly different from the true distribution. This is because, if n_{tr} is large, test input samples with large $b(x_j)$ are 'out of stock', and thus samples with small $b(x_j)$ will be chosen more frequently. Because the true distribution is generally flatter than the distribution specified by the resampling bias function $b(x_j)$, the weight $1/b(x_j)$ is too peaky and thus this could potentially lead to poor performance.

3.2 Improvement of Importance Estimation

To overcome this drawback, we propose an alternative method for estimating the importance weights. According to the theory of *finite population sampling* (Horvitz & Thompson, 1952), we have to distinguish two probabilities: the *selection* probability and the *inclusion* probability. The k-th selection probability $q_k(\boldsymbol{x}_j^{\mathrm{te}})$ is the probability that a

sample $\boldsymbol{x}_j^{\mathrm{te}}$ is selected at the k-th draw:

$$0 \leq q_k(\boldsymbol{x}_j^{\mathrm{te}}) \leq 1 \quad \text{and} \quad \sum_{j=1}^{n_{\mathrm{te}}} q_k(\boldsymbol{x}_j^{\mathrm{te}}) = 1.$$

On the other hand, the inclusion probability $\pi(\boldsymbol{x}_j^{\mathrm{te}})$ is the probability that $\boldsymbol{x}_j^{\mathrm{te}}$ is included in a training set of size n_{tr}:

$$0 \leq \pi(\boldsymbol{x}_j^{\mathrm{te}}) \leq 1 \quad \text{and} \quad \sum_{j=1}^{n_{\mathrm{te}}} \pi(\boldsymbol{x}_j^{\mathrm{te}}) = n_{\mathrm{tr}}.$$

In P-ALICE, the selection probability is set to be proportional to the resampling bias at the first draw, i.e., $q_1(\boldsymbol{x}_j^{\mathrm{te}}) \propto b(\boldsymbol{x}_j^{\mathrm{te}})$. Theoretically, the selection probability $q_k(\boldsymbol{x}_j^{\mathrm{te}})$ gradually changes as k increases, because selection probabilities of samples that have already been drawn are set to 0. However, the original P-ALICE keeps using the same resampling bias function until n_{tr} samples are gathered (if the chosen sample has already been taken, a new candidate point is kept re-drawn following the same resampling bias function until an unchosen sample is selected). Because the gap between the true selection probability and the initial resampling bias function grows as k increases, the original P-ALICE tends to be inaccurate if n_{tr} is not small.

Actually, the precise relation between $p_{\mathrm{tr}}(\boldsymbol{x}_j^{\mathrm{te}})$ and $p_{\mathrm{te}}(\boldsymbol{x}_j^{\mathrm{te}})$ is given by using the inclusion probability $\pi(\boldsymbol{x}_j^{\mathrm{te}})$ as

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

Thus, the correct importance weight $w(\boldsymbol{x}_j^{\mathrm{te}})$ is given as

$$w(\boldsymbol{x}_j^{\mathrm{te}}) \propto \frac{1}{\pi(\boldsymbol{x}_j^{\mathrm{te}})},$$

which is expected to reduce the estimation bias when n_{tr} is not small. Furthermore, this will also help reduce the estimation variance because it tends to be flatter than Eq.(11) (see Figure 2 again).

However, because the true inclusion probability $\pi(\boldsymbol{x}_j^{\mathrm{te}})$ is unknown, we numerically approximate it by the frequency of selecting each $\boldsymbol{x}_j^{\mathrm{te}}$ through Monte Carlo simulations[2]. That is, $w(\boldsymbol{x}_j^{\mathrm{te}})$ is estimated as

$$\widehat{w}(\boldsymbol{x}_j^{\mathrm{te}}) = \frac{1}{F(\boldsymbol{x}_j^{\mathrm{te}})}, \tag{12}$$

where $F(\boldsymbol{x}_j^{\mathrm{te}})$ is the frequency that $\boldsymbol{x}_j^{\mathrm{te}}$ was chosen. A pseudo code of our modified P-ALICE procedure is described in Figure 3 (see Appendix for the details of P-ALICE).

[2]Direct calculation of inclusion probabilities is known to be a hard problem (Hansen & Hurwitz, 1943; Madow, 1949; Midzuno, 1949; Horvitz & Thompson, 1952).

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 λ
  Compute $\{b_\lambda(\boldsymbol{x}_j^{\mathrm{te}})\}_{j=1}^{n_{\mathrm{te}}}$ with $b_\lambda(\boldsymbol{x}) = \left(\sum_{\ell,\ell'=1}^{t} [\widehat{\boldsymbol{U}}^{-1}]_{\ell,\ell'} \varphi_\ell(\boldsymbol{x}) \varphi_{\ell'}(\boldsymbol{x})\right)^{\lambda}$;
  Count the frequency of selecting $\{F_\lambda(\boldsymbol{x}_j^{\mathrm{te}})\}_{j=1}^{n_{\mathrm{te}}}$ through Monte Carlo simulations;
  Choose $\mathcal{X}_\lambda^{\mathrm{tr}} = \{\boldsymbol{x}_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ from $\{\boldsymbol{x}_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$ following $\{b_\lambda(\boldsymbol{x}_j^{\mathrm{te}})\}_{j=1}^{n_{\mathrm{te}}}$ without replacement;
  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 $[W_\lambda]_{i,i} = (F_\lambda(\boldsymbol{x}_i^{\mathrm{tr}}))^{-1}$;
  Compute $\boldsymbol{L}_\lambda = (\boldsymbol{X}_\lambda^\top \boldsymbol{W}_\lambda \boldsymbol{X}_\lambda)^{-1} \boldsymbol{X}_\lambda^\top \boldsymbol{W}_\lambda$;
  Compute $J_{\mathrm{W}}(\lambda) = \mathrm{tr}(\widehat{\boldsymbol{U}} \boldsymbol{L}_\lambda \boldsymbol{L}_\lambda^\top)$;
End
Compute $\widehat{\lambda} = \mathrm{argmin}_\lambda J_{\mathrm{W}}(\lambda)$;
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 $\widehat{\boldsymbol{\theta}}_{\mathrm{W}} = \boldsymbol{L}_{\widehat{\lambda}} \boldsymbol{y}^{\mathrm{tr}}$;

Figure 3: Pseudo code of the proposed algorithm.

4 Experiments

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

4.1 Toy Dataset

We first illustrate how the proposed and existing active learning methods behave under a controlled setting. Let the input dimension be $d = 1$ and let the learning target function be

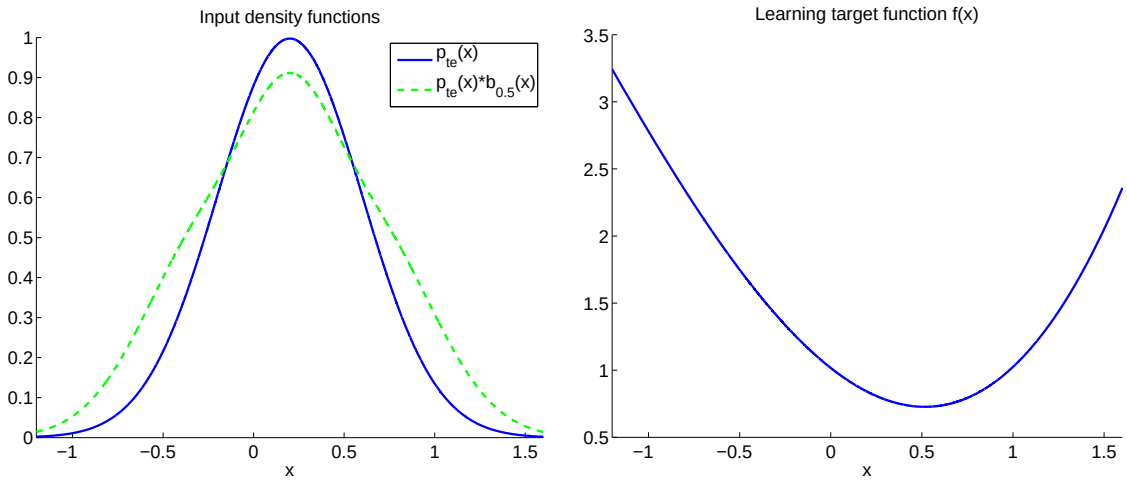
$$f(x) = 1 - x + x^2 + \delta r(x),$$

where

$$r(x) = \frac{z^3 - 3z}{\sqrt{6}} \quad \text{with} \quad z = \frac{x - 0.2}{0.4}. \tag{13}$$

We set the test input density $p_{\mathrm{te}}(x)$ to the Gaussian density with mean 0.2 and standard deviation 0.4, which is treated as unknown here. See Figure 4(a) for the profile of $p_{\mathrm{te}}(x)$. Let us construct a pool of input points by drawing $n_{\mathrm{te}} = 1000$ points independently from the test input distribution.

Let $\delta = 0.04$ 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 also treated as unknown here. See Figure 4(b) for the profile



(a) Input density functions. (b) Learning target function.

Figure 4: Input density functions and learning target function for generating toy dataset.

of $f(x)$. A polynomial model of order 2 is used for learning[3]:

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

We compare the performance of the following strategies (see Appendix for the details of each method).

P-ALICE(π^{-1})**:** Training input points are drawn following

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

for

$$\lambda \in \Lambda = \{0, 0.05, 0.10, \ldots, 1.00\}. \tag{15}$$

Then the best value of λ that minimizes J_{W} is chosen from the above candidates. IWLS (importance-weighted least-squares) is used for parameter learning. Weight functions in P-ALICE and IWLS are determined by Eq.(12), which are numerically approximated by Monte Carlo simulations with 10000 repetitions.

P-ALICE(b^{-1})**:** Training input points are drawn following Eq.(14) for Eq.(15) and the best value of λ that minimizes J_{W} is chosen. IWLS is used for parameter learning. The weight function in P-ALICE and IWLS is determined by Eq.(11).

[3]For these basis functions, the residual function $r(x)$ in Eq.(13) (which is actually a Hermite polynomial) fulfills the orthogonality condition (18) and normalization condition (19) (see Appendix for details).

P-CV$_{\mathrm{O}}$: Training input points are drawn following Eq.(14) for Eq.(15) and the best value of λ that minimizes J_{O} is chosen. OLS is used for parameter learning.

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

Figure 5 depicts the bias B, the variance V, and the generalization error G averaged over 10000 trials as functions of n_{tr}, where the model error δ^2 (that corresponds to the residual function $r(x)$) is subtracted from G for better comparison (see Appendix for details).

Figure 5(a) shows behaviors of the bias. P-CV$_{\mathrm{O}}$ has larger bias than Passive, due to the influence of a covariate shift. On the other hand, P-ALICE(b^{-1}) and P-ALICE(π^{-1}) have smaller bias than Passive, implying that their importance-weighting schemes help alleviate the influence of a covariate shift. However, as n_{tr} increases, P-ALICE(b^{-1}) tends to have larger bias than P-ALICE(π^{-1}), because importance weight estimation in P-ALICE(b^{-1}) becomes inaccurate due to the influence of sampling without replacement.

Figure 5(b) shows behaviors of the variance. All three active learning methods have smaller variance than Passive. P-CV$_{\mathrm{O}}$ has the smallest variance, and P-ALICE(π^{-1}) has smaller variance than P-ALICE(b^{-1}) particularly when n_{tr} is not small. This implies that our candidate set of resampling bias functions helps reduce the variance, and the variance tends to be smaller as the weight becomes flatter.

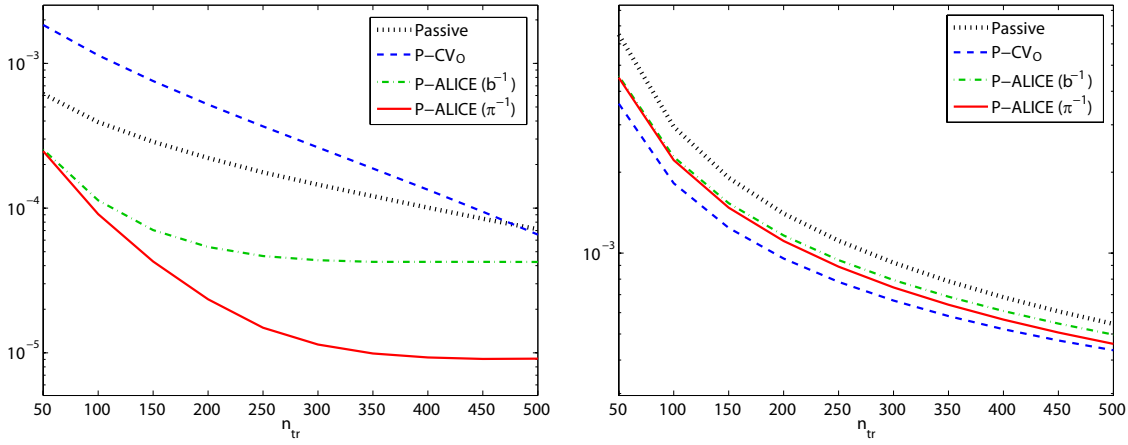
Figure 5(c) shows behaviors of the generalization error. Among all methods, P-ALICE(π^{-1}) always has the smallest generalization error. P-ALICE(b^{-1}) achieves almost the same performance as P-ALICE(π^{-1}) when n_{tr} is small, but it is outperformed by P-ALICE(π^{-1}) when n_{tr} is not small.

In Table 1, the mean squared test error obtained by each method is described for $n_{\mathrm{tr}} = 50, 200, 400$. After drawing training samples, the remaining samples in the pool may have a different density than the original test input density. Here, we computed the mean squared error not using the remaining samples in the pool, but using another 10000 test samples newly drawn from the test input density. The numbers in the table are means and standard deviations of the error over 10000 trials. In each row of the table, the best method and comparable ones by the *t-test* (Henkel, 1976) at the significance level 5% are indicated with '$\circ$'. We can see that proposed P-ALICE(π^{-1}) always performs better than other sampling schemes.

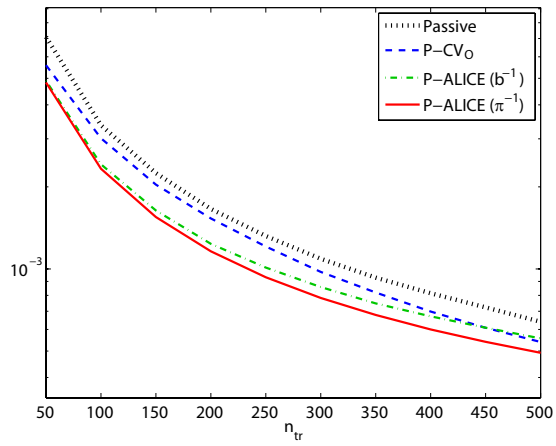
Overall, we confirmed that the importance weight plays an important role in reducing an estimation bias caused by a covariate shift, and our new weighting scheme gives further improvement upon P-ALICE; it reduces both the bias and variance when n_{tr} is not sufficiently small relative to n_{te}.

4.2 Benchmark Datasets

Finally, we compare the proposed and existing active learning methods on more challenging benchmark datasets.



(a) B (b) V



(c) G

Figure 5: The bias B, the variance V, and the generalization error G as functions of n_{tr} for the toy dataset, averaged over 10000 trials. For better comparison, the model error δ^2 is subtracted from G.

Table 1: The mean squared test error for the toy dataset (means and standard deviations over 10000 trials). All values are multiplied by 10^3 for better comparison. In each row of the table, the best method and comparable ones by the t-test at the significance level 5% are indicated with '$\circ$'.

	P-ALICE(π^{-1})	P-ALICE(b^{-1})	P-CV$_{\mathrm{O}}$	Passive
$n_{\mathrm{tr}} = 50$	$\circ$6.28±4.30	6.34±4.34	7.03±4.68	8.54±6.81
$n_{\mathrm{tr}} = 200$	$\circ$2.61±1.32	2.68±1.36	2.98±1.55	3.11±1.68
$n_{\mathrm{tr}} = 400$	$\circ$2.05±1.02	2.12±1.03	2.14±1.11	2.26±1.11

The *Bank*, *Kin*, and *Pumadyn* regression benchmark data families provided by DELVE[4] 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 $[0, 1]$. 1000 input samples are used as a pool of input points (i.e., $n_{\mathrm{te}} = 1000$), and the remaining 7192 samples are used only for performance evaluation. We draw $n_{\mathrm{tr}} = 500$ training samples from the pool. The following linear regression model is used for learning:

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

where $h = \mathrm{median}\left(\{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|\}_{i,j=1}^{n_{\mathrm{te}}}\right)$ and $\{\boldsymbol{c}_\ell\}_{\ell=1}^{50}$ are template points randomly chosen from the pool of test input points. λ is chosen from

$$\Lambda = \{0, 0.1, 0.2, \ldots, 2.0\}. \tag{16}$$

Other settings are the same as the toy experiments in Section 4.1.

In Table 2, mean squared test errors obtained by each method averaged over 100 trials are described. For better comparison, all values are normalized by the mean error of the Passive method. The best method and comparable ones by the t-test at the significance level 5% are indicated with '$\circ$'.

The table shows that P-CV$_{\mathrm{O}}$ overall performs rather well, implying that the bias tends to be dominated by the variance due to the high-dimensionality of the problems. Indeed, P-CV$_{\mathrm{O}}$ actually performs the best for some datasets. However, P-CV$_{\mathrm{O}}$ is outperformed by the baseline Passive sampling scheme for some datasets. This is probably due to the bias caused by a covariate shift. As a result, the behavior of P-CV$_{\mathrm{O}}$ tends to be unstable due to its sensitivity to model misspecification. On the other hand, the importance-weighted methods P-ALICE(b^{-1}) and P-ALICE(π^{-1}) tend to perform more reliably, which are consistently better than the baseline Passive methods. Among them, the proposed P-ALICE(π^{-1}) performs better than P-ALICE(b^{-1}).

[4] `http://www.cs.toronto.edu/~delve/`

Table 2: The mean squared test error for benchmark datasets (means and standard deviations over 100 trials). For better comparison, all values are normalized by the mean error of the Passive method. The best method and comparable ones by the t-test at the significance level 5% are indicated with '∘'.

	P-ALICE(π^{-1})	P-ALICE(b^{-1})	P-CV$_{\mathrm{O}}$	Passive
Bank-8fm	∘0.984±0.017	0.988±0.019	∘0.981±0.014	1.000±0.021
Bank-8fh	∘0.985±0.018	0.989±0.019	∘0.984±0.014	1.000±0.022
Bank-8nm	∘0.966±0.019	0.974±0.022	1.021±0.020	1.000±0.033
Bank-8nh	∘0.979±0.020	0.984±0.022	0.992±0.013	1.000±0.024
Bank-32fm	∘0.973±0.018	0.983±0.019	0.989±0.015	1.000±0.023
Bank-32fh	∘0.985±0.019	0.990±0.021	0.994±0.017	1.000±0.023
Bank-32nm	∘0.985±0.021	∘0.985±0.022	1.022±0.020	1.000±0.027
Bank-32nh	∘0.990±0.016	0.993±0.017	0.999±0.015	1.000±0.019
Kin-8fm	∘0.990±0.017	∘0.992±0.017	1.000±0.016	1.000±0.020
Kin-8fh	0.989±0.014	0.994±0.015	∘0.977±0.014	1.000±0.018
Kin-8nm	∘0.988±0.017	0.994±0.018	0.991±0.014	1.000±0.019
Kin-8nh	∘0.991±0.018	0.995±0.019	∘0.990±0.016	1.000±0.020
Kin-32fm	∘0.990±0.014	0.994±0.017	1.006±0.017	1.000±0.017
Kin-32fh	0.993±0.014	0.995±0.015	∘0.989±0.012	1.000±0.016
Kin-32nm	0.995±0.015	0.997±0.014	∘0.992±0.013	1.000±0.014
Kin-32nh	0.994±0.014	0.997±0.015	∘0.991±0.011	1.000±0.015
Pumadyn-8fm	0.990±0.022	0.994±0.022	∘0.979±0.017	1.000±0.024
Pumadyn-8fh	0.994±0.018	0.995±0.018	∘0.986±0.015	1.000±0.019
Pumadyn-8nm	0.985±0.014	0.993±0.017	∘0.981±0.013	1.000±0.019
Pumadyn-8nh	∘0.989±0.018	0.992±0.018	1.007±0.014	1.000±0.019
Pumadyn-32fm	∘0.982±0.015	0.989±0.016	0.993±0.016	1.000±0.018
Pumadyn-32fh	∘0.991±0.012	0.995±0.014	∘0.990±0.012	1.000±0.015
Pumadyn-32nm	0.991±0.021	0.997±0.022	∘0.988±0.019	1.000±0.022
Pumadyn-32nh	0.992±0.015	0.995±0.016	∘0.988±0.013	1.000±0.016
Average	∘0.988±0.019	0.992±0.019	0.994±0.019	1.000±0.021

5 Conclusions

In this paper, we discussed importance weight estimation in the pool-based batch active learning criterion called P-ALICE. We pointed out that when the number of training samples to gather is not small compared with the pool size, importance weights used in the original P-ALICE are not accurate. This inaccuracy is due to the influence of sampling without replacement.

To cope with this problem, we proposed an alternative method of importance weight estimation based on the inclusion probability. Because the true inclusion probability is generally unknown, we numerically approximated it by the frequency of selection of each

sample through Monte Carlo simulations.

The importance weights obtained by the proposed approach is more accurate when the sampling rate is not small, and thus it achieves a lower estimation bias. Furthermore, because the importance weights obtained by the proposed approach tends to be flatter than the original ones, it also reduces the variance. Numerical experiments with toy and benchmark datasets showed that our new weighting scheme gave statistically significant improvement upon the original P-ALICE.

The importance of active learning research grows significantly in recent years because labeling costs became a critical bottleneck of real-world machine learning applications. In consideration of this increasing interest and demand in active learning, further enhancing the active learning performance is an important challenge, for instance, in the context of *crowdsourcing*.

Acknowledgments

MS was supported by MEXT KAKENHI 23120004.

References

Akiyama, T., Hachiya, H., & Sugiyama, M. (2010). Efficient exploration through active learning for value function approximation in reinforcement learning. *Neural Networks*, *23*, 639–648.

Beygelzimer, A., Dasgupta, S., & Langford, J. (2009). Importance weighted active learning. *Proceedings of the 26th Annual International Conference on Machine Learning* (pp. 49–56).

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer.

Box, G. E. P., & Hunter, W. G. (1965). Sequential design of experiments for nonlinear models. *Proceedings of IBM Scientific Computing Symposium in Statistics* (pp. 113–137).

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

Fedorov, V. V. (1972). *Theory of Optimal Experiments*. New York, NY, USA: Academic Press.

Fishman, G. S. (1996). *Monte Carlo: Concepts, Algorithms, and Applications*. Berlin, Germany: Springer-Verlag.

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

Hansen, M. H., & Hurwitz, W. N. (1943). On the theory of sampling from finite populations. *The Annals of Mathematical Statistics*, *14*, 333–362.

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York, NY, USA: Springer.

Henkel, R. E. (1976). *Tests of Significance*. Beverly Hills, CA, USA.: SAGE Publication.

Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, *12*, 55–67.

Horvitz, D. G., & Thompson, D. J. (1952). A generalization of sampling without replacement from a finite universe. *Journal of the American Statistical Association*, *47*, 663–685.

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

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

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

Lewis, D., & Gale, W. (1994). A sequential algorithm for training text classifiers. *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 3–12).

Liu, Y. (2004). Active learning with support vector machine applied to gene expression data for cancer classification. *Journal of Chemical Information and Computer Sciences*, *44*, 1936–1941.

Madow, W. G. (1949). On the theory of systematic sampling, II. *Annals of Mathematical Statistics*, *20*, 333–354.

McCallum, A., & Nigam, K. (1998). Employing EM in pool-based active learning for text classification. *Proceedings of the 15th International Conference on Machine Learning*.

Midzuno, H. (1949). An outline of the theory of sampling systems. *Annals of the Institute of Statistical Mathematics*, *1*, 149–156.

Poggio, T., & Girosi, F. (1990). Networks for approximation and learning. *Proceedings of the IEEE*, *78*, 1481–1497.

Pukelsheim, F. (1993). *Optimal Design of Experiments*. New York, NY, USA: Wiley.

Quiñonero-Candela, J., Sugiyama, M., Schwaighofer, A., & Lawrence, N. (Eds.). (2009). *Dataset Shift in Machine Learning*. Cambridge, MA, USA: MIT Press.

Settles, B. (2009). *Active Learning Literature Survey*, Computer Sciences Technical Report 1648. University of Wisconsin–Madison.

Shimodaira, H. (2000). Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, *90*, 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., & Kawanabe, M. (2012). *Machine Learning in Non-Stationary Environments: Introduction to Covariate Shift Adaptation*. Cambridge, MA, USA: MIT Press.

Sugiyama, M., & Nakajima, S. (2009). Pool-based active learning in approximate linear regression. *Machine Learning*, *75*, 249–274.

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

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

Sugiyama, M., & Rubens, N. (2008). A batch ensemble approach to active learning with model selection. *Neural Networks*, *21*, 1278–1286.

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 and N. Lawrence (Eds.), *Dataset Shift in Machine Learning*, chapter 7, 107–130. Cambridge, MA, USA: MIT Press.

Tikhonov, A. N., & Arsenin, V. Y. (1977). *Solutions of Ill-Posed Problems*. Washington, DC, USA: V. H. Winston.

Ueki, K., Sugiyama, M., & Ihara, Y. (2010). A semi-supervised approach to perceived age prediction from face images. *IEICE Transactions on Information and Systems*, *E93-D*, 2875–2878.

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*, 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*, 395–412.

Wiens, J., & Guttag, J. (2010). Active learning applied to patient-adaptive heartbeat classification. In J. Lafferty, C. K. I. Williams, R. Zemel, J. Shawe-Taylor and A. Culotta (Eds.), *Advances in Neural Information Processing Systems 23*, 2442–2450.

A Review of Existing Active Learning Methods

Here, we describe a basic strategy for population-based active learning and review existing methods and their extensions to pool-based scenarios.

A.1 Basic Strategy for Population-Based Active Learning

Population-based active learning is the problem of optimally designing the training input density p_{tr} so that the generalization error (9) is minimized:

$$\min_{p_{\mathrm{tr}}} G(p_{\mathrm{tr}}).$$

Thus, the generalization error needs to be reasonably estimated as a functional of p_{tr} before observing output values.

Let us decompose that the learning target function $f(\boldsymbol{x})$ as

$$f(\boldsymbol{x}) = g(\boldsymbol{x}) + \delta r(\boldsymbol{x}), \tag{17}$$

where $g(\boldsymbol{x})$ is the optimal approximation to $f(\boldsymbol{x})$ by the model (1):

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

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

$$\boldsymbol{\theta}^* = \operatorname*{argmin}_{\boldsymbol{\theta}} G.$$

$\delta r(\boldsymbol{x})$ in Eq.(17) is the residual function, which is orthogonal to $\{\varphi_\ell(\boldsymbol{x})\}_{\ell=1}^t$ under $p_{\mathrm{te}}(\boldsymbol{x})$ (see Figure 6):

$$\int r(\boldsymbol{x}^{\mathrm{te}}) \varphi_\ell(\boldsymbol{x}^{\mathrm{te}}) p_{\mathrm{te}}(\boldsymbol{x}^{\mathrm{te}}) \mathrm{d}\boldsymbol{x}^{\mathrm{te}} = 0 \qquad \text{for } \ell = 1, 2, \ldots, t. \tag{18}$$

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

$$\int \left(r(\boldsymbol{x}^{\mathrm{te}})\right)^2 p_{\mathrm{te}}(\boldsymbol{x}^{\mathrm{te}}) \mathrm{d}\boldsymbol{x}^{\mathrm{te}} = 1. \tag{19}$$

Therefore, a scalar δ corresponds to the degree of model misspecification. In traditional active learning literature (Kiefer, 1959; Fedorov, 1972; Cohn et al., 1996; Fukumizu, 2000),

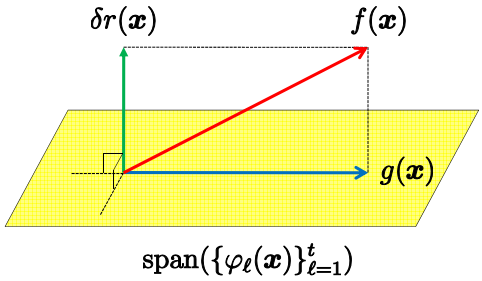


Figure 6: Orthogonal decomposition of $f(\boldsymbol{x}^{\mathrm{tr}})$.

the model is assumed to be correctly specified, i.e., δ is exactly zero. However, such a strict model assumption may not be satisfied in practice. In this paper, we assume that the model is *approximately* correct, i.e., sufficiently small δ is still allowed[5].

To obtain a generalization error estimator, we analyze the *conditional*-expectation of the generalization error, i.e., the expectation over the training output noise $\{\epsilon_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$, given the realization of training input points $\{\boldsymbol{x}^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ (Sugiyama et al., 2009). Let $\mathbb{E}_{\boldsymbol{\epsilon}}$ 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 three terms:

$$\mathop{\mathbb{E}}_{\boldsymbol{\epsilon}} G = B + V + \delta^2, \tag{20}$$

where

$$B = \int \left(\mathop{\mathbb{E}}_{\boldsymbol{\epsilon}} \widehat{f}(\boldsymbol{x}^{\mathrm{te}}) - g(\boldsymbol{x}^{\mathrm{te}}) \right)^2 p_{\mathrm{te}}(\boldsymbol{x}^{\mathrm{te}}) \mathrm{d}\boldsymbol{x}^{\mathrm{te}},$$
$$V = \int \mathop{\mathbb{E}}_{\boldsymbol{\epsilon}} \left(\widehat{f}(\boldsymbol{x}^{\mathrm{te}}) - \mathop{\mathbb{E}}_{\boldsymbol{\epsilon}} \widehat{f}(\boldsymbol{x}^{\mathrm{te}}) \right)^2 p_{\mathrm{te}}(\boldsymbol{x}^{\mathrm{te}}) \mathrm{d}\boldsymbol{x}^{\mathrm{te}}.$$

The first term B in Eq.(20) corresponds to the (squared) *bias*. This term is not accessible without observing $\{y_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ because it contains the unknown function $g(\boldsymbol{x})$. The second term V in Eq.(20) corresponds to the *variance*. In the current setup, V can be expressed as

$$V = \sigma^2 \mathrm{tr}(\boldsymbol{U} \boldsymbol{L}_{\mathrm{W}} \boldsymbol{L}_{\mathrm{W}}^\top),$$

where σ^2 is the noise variance and $\boldsymbol{U}$ is the $t \times t$ matrix with the (ℓ, ℓ')-th element

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

Thus, we can access this term without observing $\{y_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ except the scaling factor σ^2. The third term δ^2 in Eq.(20) corresponds to the model error. This term can be safely

[5]When δ is large, learning itself may not work well with such a strongly misspecified model. In such a case, model selection needs to be performed (Sugiyama & Rubens, 2008).

ignored in the minimization problem because δ is a constant that depends neither on $p_{\rm tr}$ nor $\{\boldsymbol{x}_i^{\rm tr}\}_{i=1}^{n_{\rm tr}}$.

Below, we consider an approximate regression model, which causes a bias but the bias is dominated by the variance. Therefore, our basic strategy for active learning is the *variance-only* approach, in which the generalization error is estimated only from the variance term[6].

A.2 Conditional-Expectation Variance-Only Active Learning for IWLS: ALICE

We first review a population-based batch active learning criterion called ALICE (*Active Learning using Importance-weighted least-squares learning based on Conditional Expectation of the generalization error*) following Sugiyama (2006).

In ALICE, the *importance weight* (Fishman, 1996), which refers to the ratio of the test and training input densities,

$$w(\boldsymbol{x}) = \frac{p_{\rm te}(\boldsymbol{x})}{p_{\rm tr}(\boldsymbol{x})}, \tag{21}$$

is used as the weight function in WLS. This method is particularly called *Importance-Weighted Least-Squares (IWLS)*.

Let $G_{\rm W}$, $B_{\rm W}$, and $V_{\rm W}$ be G, B, and V for a learned function obtained by IWLS, respectively. Then it was shown that, for IWLS with an approximately correct model, B and V are expressed as

$$\begin{aligned} B_{\rm W} &= \mathcal{O}_p(\delta^2 n_{\rm tr}^{-1}), \\ V_{\rm W} &= \sigma^2 {\rm tr}(\boldsymbol{U}\boldsymbol{L}_{\rm W}\boldsymbol{L}_{\rm W}^\top) = \mathcal{O}_p(n_{\rm tr}^{-1}). \end{aligned}$$

Note that the asymptotic orders in the above equations are in probability because random variables $\{\boldsymbol{x}_i^{\rm tr}\}_{i=1}^{n_{\rm tr}}$ are included. These equations imply that if $\delta = o_p(1)$,

$$\mathop{\mathbb{E}}_{\boldsymbol{\epsilon}} G_{\rm W} = \sigma^2 {\rm tr}(\boldsymbol{U}\boldsymbol{L}_{\rm W}\boldsymbol{L}_{\rm W}^\top) + o_p(n_{\rm tr}^{-1}).$$

Motivated by this asymptotic form, the active learning criterion of ALICE is given as

$$J_{\rm W} = {\rm tr}(\boldsymbol{U}\boldsymbol{L}_{\rm W}\boldsymbol{L}_{\rm W}^\top). \tag{22}$$

Based on this estimator, the optimal training input density $p_{\rm tr}$ is searched from the set $\mathcal{P}$ of all strictly positive probability densities as

$$p_{\rm tr}^{\rm ALICE} = \mathop{\rm argmin}_{p_{\rm tr}\in\mathcal{P}} J_{\rm W}.$$

[6]Another possibility is the *bias-and-variance* approach, in which the bias term is also estimated using a small number of training samples chosen randomly in the beginning (Kanamori & Shimodaira, 2003). However, the variance-only approach appears to be practically more useful for approximate regression models because bias estimation is difficult (Sugiyama, 2006).

Practically, $\mathcal{P}$ may be replaced by a finite subset $\widehat{\mathcal{P}}$ and choose the one that minimizes J_{W} from the set $\widehat{\mathcal{P}}$. A useful heuristic for determining $\widehat{\mathcal{P}}$ was also proposed in Sugiyama and Nakajima (2009), which will be explained later.

A.3 Conditional-Expectation Variance-Only Active Learning for OLS: $\mathrm{CV_O}$

If we use the uniform weight in WLS, or equivalently if we set $\boldsymbol{W}$ defined by Eq.(4) to be the $n_{\mathrm{tr}} \times n_{\mathrm{tr}}$ identity matrix, WLS is reduced to *Ordinary Least-Squares (OLS)*:

$$\widehat{\boldsymbol{\theta}}_{\mathrm{O}} = \boldsymbol{L}_{\mathrm{O}} \boldsymbol{y}^{\mathrm{tr}}, \tag{23}$$

where

$$\boldsymbol{L}_{\mathrm{O}} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top. \tag{24}$$

The subscript 'O' denotes 'Ordinary'. Here, we review a population-based active learning method based on OLS (Fedorov, 1972; Cohn et al., 1996; Fukumizu, 2000), which we refer to as *CV_O (Conditional-expectation Variance-only active learning for OLS)*.

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 (Sugiyama, 2006; Sugiyama & Nakajima, 2009):

$$\begin{aligned} B_{\mathrm{O}} &= \mathcal{O}(\delta^2), \\ V_{\mathrm{O}} &= \sigma^2 \mathrm{tr}(\boldsymbol{U} \boldsymbol{L}_{\mathrm{O}} \boldsymbol{L}_{\mathrm{O}}^\top) = \mathcal{O}_p(n_{\mathrm{tr}}^{-1}). \end{aligned}$$

The above equations imply that if $\delta = o_p(n_{\mathrm{tr}}^{-\frac{1}{2}})$,

$$\mathop{\mathbb{E}}_{\boldsymbol{\epsilon}} G_{\mathrm{O}} = \sigma^2 \mathrm{tr}(\boldsymbol{U} \boldsymbol{L}_{\mathrm{O}} \boldsymbol{L}_{\mathrm{O}}^\top) + o_p(n_{\mathrm{tr}}^{-1}).$$

Motivated by this asymptotic form, the active learning criterion of $\mathrm{CV_O}$ is given as

$$J_{\mathrm{O}} = \mathrm{tr}(\boldsymbol{U} \boldsymbol{L}_{\mathrm{O}} \boldsymbol{L}_{\mathrm{O}}^\top). \tag{25}$$

Then the optimal training input density of $\mathrm{CV_O}$ is given as

$$p_{\mathrm{tr}}^{\mathrm{CV_O}} = \mathop{\mathrm{argmin}}_{p_{\mathrm{tr}} \in \mathcal{P}} J_{\mathrm{O}}.$$

A.4 ALICE vs. $\mathrm{CV_O}$

In active learning scenarios, the training input density generally differs from the test input density, which is called a *covariate shift* (Shimodaira, 2000; Quiñonero-Candela et al., 2009; Sugiyama & Kawanabe, 2012). It is known that a misspecified model under a covariate shift yields a significant estimation bias. Indeed, OLS with a misspecified model

is not unbiased even asymptotically under a covariate shift. On the other hand, IWLS is asymptotically unbiased thanks to the importance weight, which could be intuitively understood by the following identity (Fishman, 1996):

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

where $w(\boldsymbol{x})$ in the above equation is the importance weight (21). This is the reason why the model requirement for ALICE ($\delta = o_p(1)$) is weaker than that for $\mathrm{CV_O}$ ($\delta = o_p(n_{\mathrm{tr}}^{-\frac{1}{2}})$). This implies that ALICE has a wider range of applications than $\mathrm{CV_O}$, and thus is more promising in practical situations where the model at hand is not correctly specified.

A.5 Full-Expectation Variance-Only Active Learning for IWLS: $\mathrm{FV_W}$

Next, we review a population-based active learning criterion that we refer to as *Full-expectation Variance-only active learning for WLS (FV_W)*. For IWLS, Kanamori and Shimodaira (2003) proved that the full-expectation of the generalization error, i.e., the expectation over both training input points $\{\boldsymbol{x}_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ and training output noise $\{\epsilon_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ is asymptotically expressed as

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

where $\mathbb{E}_{\boldsymbol{x}}$ 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

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

where $w(\boldsymbol{x})$ is the importance weight (21). Note that $\frac{1}{n_{\mathrm{tr}}}\mathrm{tr}(\boldsymbol{U}^{-1}\boldsymbol{S})$ corresponds to the squared bias, whereas $\frac{\sigma^2}{n_{\mathrm{tr}}}\mathrm{tr}(\boldsymbol{U}^{-1}\boldsymbol{T})$ corresponds to the variance.

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

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

Based on this asymptotic form, the criterion of $\mathrm{FV_W}$ is given as follows (Wiens, 2000):

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

where

$$J_{\mathrm{FW}} = \frac{1}{n_{\mathrm{tr}}}\mathrm{tr}(\boldsymbol{U}^{-1}\boldsymbol{T}). \tag{26}$$

The subscript 'FW' denotes 'Full' and 'Weighted'. A notable property of $\mathrm{FV_W}$ is that the optimal training input density $p_{\mathrm{tr}}^{\mathrm{FV_W}}$ can be obtained in a closed form (Wiens, 2000; Kanamori, 2007):

$$p_{\mathrm{tr}}^{\mathrm{FV_W}}(\boldsymbol{x}) \propto p_{\mathrm{te}}(\boldsymbol{x}) b_{\mathrm{FV_W}}(\boldsymbol{x}), \tag{27}$$

where

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

Eq.(27) implies that the importance weight for the optimal training input density $p_{\mathrm{tr}}^{\mathrm{FV_W}}(\boldsymbol{x})$ is given by

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

A.6 ALICE vs. $\mathrm{FV_W}$

As we have mentioned, there are two ways to take an expectation of the generalization error: *full*-expectation (i.e., the generalization error is expected over *both* training output noise and training input realization) and *conditional*-expectation (i.e., the generalization error is expected over training output noise *given* training input realization).

Our ideal approach is to directly evaluate 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, such single-trial evaluation is not possible in practice because we are not allowed to access realized values of the noise $\{\epsilon_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$. Thus, taking an expectation over the noise may not be avoidable. On the other hand, the location of training input points $\{\boldsymbol{x}_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ is accessible by nature, and utilizing this information may help us estimate the single-trial generalization error accurately. Therefore, the conditional-expectation (or input-dependent) approach is more promising for the single-trial analysis than the full-expectation (or input-independent) approach.

Motivated by this idea, ALICE adopts the generalization error estimator J_{W} that is based on the conditional-expectation analysis. On the other hand, the estimator J_{FW} used in $\mathrm{FV_W}$ is based on the full-expectation analysis. Sugiyama (2006) showed that J_{W} (22) and J_{FW} (26) are related to each other by

$$J_{\mathrm{W}} = J_{\mathrm{FW}} + \mathcal{O}_p(n_{\mathrm{tr}}^{-\frac{3}{2}}),$$

which implies that they are actually equivalent asymptotically. However, they are still 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:

$$\mathop{\mathbb{E}}_{\boldsymbol{\epsilon}}(\sigma^2 J_{\mathrm{FW}} - G_{\mathrm{W}})^2 \geq \mathop{\mathbb{E}}_{\boldsymbol{\epsilon}}(\sigma^2 J_{\mathrm{W}} - G_{\mathrm{W}})^2.$$

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

We have explained that ALICE is more preferable than FV_{W} for the single-trial generalization error analysis. However, although FV_{W} has an analytic solution, the minimizer of J_{W} cannot be obtained in a closed form due to its input-dependent nature. Thus, an efficient strategy is necessary for searching the ALICE solution. A useful heuristic is to search the solution around the minimizer of J_{FW}. We will explain this in more detail below.

A.7 Extension to Pool-based Scenarios

In this section, we review extensions of ALICE, CV_{O}, and FV_{W} to pool-based scenarios (Sugiyama & Nakajima, 2009), which we call P-ALICE, P-CV_{O}, and P-FV_{W}, respectively.

In the pool-based setting, we are given a 'pool' of test input points $\{\boldsymbol{x}_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$ drawn independently from $p_{\mathrm{te}}(\boldsymbol{x})$ (> 0 for all $\boldsymbol{x} \in \mathcal{D}$). From the pool, we choose n_{tr} training input points for observing output values. The objective of pool-based active learning is to choose training input points $\{\boldsymbol{x}_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ from a pool of test input points $\{\boldsymbol{x}_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$ so that the generalization error is minimized.

In the pool-based scenario, we are not allowed to directly access $p_{\mathrm{te}}(\boldsymbol{x})$. Moreover, possible locations of training input points are not arbitrary, but limited to choosing from the test pool. This implies that directly determining a training input density itself and drawing i.i.d. training input points from the training input distribution is not possible. Instead, we are only allowed to choose training input points from the test pool and thus $p_{\mathrm{tr}}(\boldsymbol{x})$ is also unknown. These limitations cause the following three issues:

(a) The expectation over $p_{\mathrm{te}}(\boldsymbol{x})$ contained in $\boldsymbol{U}$ is not directly computable.

(b) A sampling strategy to gather training input points $\{\boldsymbol{x}_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ from the test pool $\{\boldsymbol{x}_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$ is necessary.

(c) The importance weight $p_{\mathrm{te}}(\boldsymbol{x})/p_{\mathrm{tr}}(\boldsymbol{x})$ at training input points $\{\boldsymbol{x}_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ contained in $\boldsymbol{L}_{\mathrm{W}}$ through $\boldsymbol{W}$ is not directly computable.

Regarding (a), we practically approximate the expectation over $p_{\mathrm{te}}(\boldsymbol{x})$ by the expectation over test input samples $\{\boldsymbol{x}_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$, which is consistent. More specifically, $\boldsymbol{U}$ in Eqs.(22), (25), and (26) is replaced with its empirical estimate $\widehat{\boldsymbol{U}}$ defined by Eq.(10).

Regarding (b), Sugiyama and Nakajima (2009) employed a *resampling bias function* $b(\boldsymbol{x})$ defined over the pool of samples to draw training samples, i.e., a training input point is chosen one by one with probability proportional to the resampling bias function; then the selection weight for the chosen sample is set to zero to prevent overlapping.

Regarding (c), Sugiyama and Nakajima (2009) proposed a simple estimation scheme. The above sampling procedure implies that if $n_{\mathrm{tr}} \ll n_{\mathrm{te}}$, the training input density $p_{\mathrm{tr}}(\boldsymbol{x}_j^{\mathrm{te}})$ is regarded as being proportional to the product of the test input density $p_{\mathrm{te}}(\boldsymbol{x}_j^{\mathrm{te}})$ and the

resampling bias function $b(\boldsymbol{x}_j^{\mathrm{te}})$, i.e.,

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

Then the importance weight $w(\boldsymbol{x}_j^{\mathrm{te}})$ is given as

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

In practice, users prepare a candidate set of $b(\boldsymbol{x})$ and choose the best one based on their generalization error estimators ($J_{\mathrm{W}} = \mathrm{tr}(\widehat{\boldsymbol{U}}\boldsymbol{L}_{\mathrm{W}}\boldsymbol{L}_{\mathrm{W}}^{\top})$ for P-ALICE and $J_{\mathrm{O}} = \mathrm{tr}(\widehat{\boldsymbol{U}}\boldsymbol{L}_{\mathrm{O}}\boldsymbol{L}_{\mathrm{O}}^{\top})$ for P-CV$_{\mathrm{O}}$). Therefore, to achieve a good learning performance, we need to prepare reasonable candidates of $b(\boldsymbol{x})$. Sugiyama and Nakajima (2009) proposed to use the closed-form solution of P-FV$_{\mathrm{W}}$ as a 'base' of the candidates and search for the best solution around its vicinity. More specifically, the following family of resampling bias functions parameterized by a scalar λ is used as a candidate set:

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

The parameter λ controls the 'shape' of the training input distribution. The optimal solution of P-FV$_{\mathrm{W}}$ corresponds to $\lambda = 1/2$, and *passive learning* (i.e., the training and test input densities are equivalent) corresponds to $\lambda = 0$. Practically, the best value of λ may be searched for by simple multi-point search, i.e., the value of J_{W} is computed for several different values of λ and the minimizer is chosen.