Active Learning with Model Selection in Linear Regression*

Masashi Sugiyama and Neil Rubens

Department of Computer Science,
Tokyo Institute of Technology, Tokyo, Japan.
sugi@cs.titech.ac.jp

Abstract

Optimally designing the location of training input points (active learning) and choosing the best model (model selection) are two important components of supervised learning and have been studied extensively. However, these two issues seem to have been investigated separately as two independent problems. If training input points and models are simultaneously optimized, the generalization performance would be further improved. In this paper, we propose a new approach called *ensemble active learning* for solving the problems of active learning and model selection at the same time. We demonstrate by numerical experiments that the proposed method compares favorably with alternative approaches such as iteratively performing active learning and model selection in a sequential manner.

Keywords

active learning, model selection, generalization error, regression. covariate shift, importance sampling, batch learning, sequential learning

1 Introduction

When we are allowed to choose the location of training input points in supervised learning, we want to optimize them so that the generalization error is minimized. This problem is called *active learning* (AL) and has been studied extensively [5, 11, 3, 6, 21, 9, 17]. On the other hand, *model selection* (MS) is another important issue in supervised learning: a model (e.g., the type and number of basis functions, regularization parameter, etc.) is optimized so that the generalization error is minimized [1, 14, 15, 4, 16, 19].

Although AL and MS share a common goal of minimizing the generalization error, they seem to have been studied separately as two independent problems. If AL and MS are performed at the same time, the generalization performance would be further improved. We call the problem of simultaneously optimizing training input points and models *active learning with model selection* (ALMS). However, ALMS can not be directly solved by simply combining standard AL methods and MS methods in a batch manner due to the AL/MS dilemma: In order to select training input points by an existing AL method, a model must be fixed (i.e., MS has been performed). On the other hand, in order to select the model by a standard MS method, the training input points must be fixed and corresponding training output values must be gathered (i.e., AL has been performed).

A standard approach to coping with the AL/MS dilemma is the *sequential approach*, i.e., iteratively performing AL and MS in an online manner [10]. Although this approach is intuitive, it can perform poorly due to the *model drift*—the chosen model varies through the online learning process. Since the location of optimal training input points depends on the model, the training input points chosen in early stages could be less useful for the model selected in the end of the learning process.

An alternative approach to solving the ALMS problems is to choose all the training input points for an initially chosen model, which we refer to as the *batch approach*. Since the target model is fixed through the learning process, this approach does not suffer from the model drift and it works optimally (in terms of AL) if the initially chosen model agrees with the finally chosen model. However, choosing an appropriate initial model *before* having any single training sample may not be possible without prior knowledge—which is usually unavailable. For this reason, it may be difficult to obtain a good performance by the batch approach.

The weakness of the batch approaches actually lies in the fact that the training input points chosen by an AL method are *overfitted* to the initially chosen model; the training input points optimized for the initial model could be poor if a different model is chosen later.

*The authors acknowledge financial support from MEXT (17700142, 18300057, and 053155), the Okawa Foundation, the Microsoft CORE3 Project, and the IBM Faculty Award.

To alleviate this problem, we propose a new approach called *ensemble active learning* (EAL). The main idea of EAL is to perform AL not for a *single* model, but for *all* models at hand. This allows us to hedge the risk of overfitting to a single (possibly inferior) model. We experimentally show the EAL method significantly outperforms other approaches.

2 Problem formulation

In this section, we formulate the supervised learning problem.

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

$$\{(\boldsymbol{x}_i, y_i) \mid y_i = f(\boldsymbol{x}_i) + \epsilon_i\}_{i=1}^{n}, \tag{2.1}$$

where d is the dimension of the input vector $\boldsymbol{x}$, n is the number of training samples, and $\{\epsilon_i\}_{i=1}^{n}$ are i.i.d. noise with mean zero and unknown variance σ^2 (see Figure 1). We draw the training input points $\{\boldsymbol{x}_i\}_{i=1}^{n}$ from a distribution with density $p(\boldsymbol{x})$, which we would like to optimize by an AL method.

We employ the following linear regression model for learning:

$$\widehat{f}(\boldsymbol{x}) = \sum_{i=1}^{b} \alpha_i \varphi_i(\boldsymbol{x}), \tag{2.2}$$

where $\{\varphi_i(\boldsymbol{x})\}_{i=1}^{b}$ are fixed linearly independent functions and $\{\alpha_i\}_{i=1}^{b}$ are parameters to be learned. Let

$$\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_b)^\top, \tag{2.3}$$

where $^\top$ denotes the transpose of a vector/matrix. We define the generalization error G of a learned function $\widehat{f}(\boldsymbol{x})$ by the expected squared error for *test* input points. We assume that the test input points are drawn independently from a distribution with density $q(\boldsymbol{x})$. Then G is expressed as

$$G = \int_{\mathcal{D}} \left(\widehat{f}(\boldsymbol{x}) - f(\boldsymbol{x})\right)^2 q(\boldsymbol{x}) d\boldsymbol{x}. \tag{2.4}$$

As in the AL literature [6, 21, 9, 17], we assume that $q(\boldsymbol{x})$ is known or its reasonable estimate is available.

Since we discuss the MS problem, i.e., choosing the number and type of the basis functions $\{\varphi_i(\boldsymbol{x})\}_{i=1}^{b}$, we can not generally assume that the model is correctly specified. Thus, the target function $f(\boldsymbol{x})$ is not necessarily of the form (2.2), but is expressed as follows (see Figure 2):

$$f(\boldsymbol{x}) = g(\boldsymbol{x}) + \delta r(\boldsymbol{x}), \tag{2.5}$$

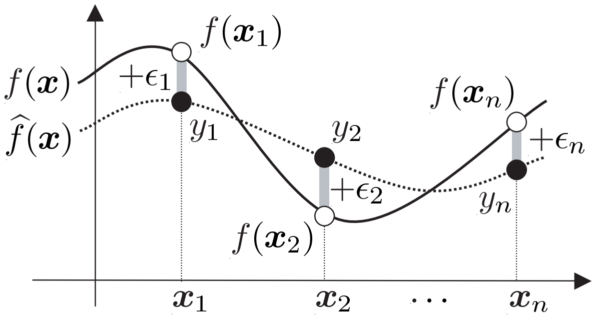


Figure 1: Regression problem.

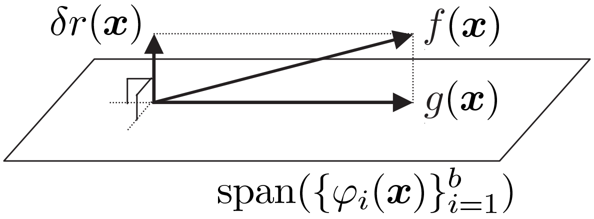

$\delta r(\boldsymbol{x})$
$f(\boldsymbol{x})$
$g(\boldsymbol{x})$
$\mathrm{span}(\{\varphi_i(\boldsymbol{x})\}_{i=1}^{b})$

Figure 2: Orthogonal decomposition of $f(\boldsymbol{x})$.

where $g(\boldsymbol{x})$ is the optimal approximation to $f(\boldsymbol{x})$ within the model (2.2):

$$g(\boldsymbol{x}) = \sum_{i=1}^{b} \alpha_i^* \varphi_i(\boldsymbol{x}). \tag{2.6}$$

$\{\alpha_i^*\}_{i=1}^{b}$ are the unknown optimal parameter under G:

$$\boldsymbol{\alpha}^* = (\alpha_1^*, \alpha_2^*, \ldots, \alpha_b^*)^\top = \underset{\boldsymbol{\alpha}}{\mathrm{argmin}}\, G. \tag{2.7}$$

$\delta r(\boldsymbol{x})$ in Eq.(2.5) is the residual, which is orthogonal to $\{\varphi_i(\boldsymbol{x})\}_{i=1}^{b}$ under $q(\boldsymbol{x})$, i.e., for $i = 1, 2, \ldots, b$,

$$\int_{\mathcal{D}} r(\boldsymbol{x}) \varphi_i(\boldsymbol{x}) q(\boldsymbol{x}) d\boldsymbol{x} = 0. \tag{2.8}$$

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

$$\int r^2(\boldsymbol{x}) q(\boldsymbol{x}) d\boldsymbol{x} = 1. \tag{2.9}$$

Under this setting, the expected generalization error can be decomposed into the model error δ^2, bias B, and variance V:

$$\mathop{\mathbb{E}}_{\boldsymbol{\epsilon}} G = \delta^2 + B + V, \tag{2.10}$$

where $\mathbb{E}_{\boldsymbol{\epsilon}}$ denotes the expectation over the noise $\{\epsilon_i\}_{i=1}^n$ and

$$B = \int_{\mathcal{D}} \left(\mathop{\mathbb{E}}_{\boldsymbol{\epsilon}} \widehat{f}(\boldsymbol{x}) - g(\boldsymbol{x}) \right)^2 q(\boldsymbol{x}) d\boldsymbol{x}, \tag{2.11}$$

$$V = \mathop{\mathbb{E}}_{\boldsymbol{\epsilon}} \int_{\mathcal{D}} \left(\widehat{f}(\boldsymbol{x}) - \mathop{\mathbb{E}}_{\boldsymbol{\epsilon}} \widehat{f}(\boldsymbol{x}) \right)^2 q(\boldsymbol{x}) d\boldsymbol{x}. \tag{2.12}$$

2.2 Parameter Learning A standard parameter learning method in the regression scenario would be ordinary least squares (OLS). OLS is asymptotically unbiased and efficient in standard cases. However, the AL scenario is a typical case of the *covariate shift* [16]—the training input distribution is different from the test input distribution:

$$p(\boldsymbol{x}) \neq q(\boldsymbol{x}). \tag{2.13}$$

Under the covariate shift, OLS is not asymptotically unbiased anymore; instead, the following *adaptive importance-weighted least-squares* (AIWLS)[1] is shown to work well [16]:

$$\min_{\boldsymbol{\alpha}} \left[\sum_{i=1}^n \left(\frac{q(\boldsymbol{x}_i)}{p(\boldsymbol{x}_i)} \right)^\lambda \left(\widehat{f}(\boldsymbol{x}_i) - y_i \right)^2 \right], \tag{2.14}$$

where λ ($0 \leq \lambda \leq 1$) is a tuning parameter.

AIWLS has the following property: When $\lambda = 0$, AIWLS is reduced to OLS; thus it is biased but has smaller variance. When $\lambda = 1$, AIWLS is asymptotically unbiased but has a larger variance. In practice, an intermediate λ often produces good results since it can control the trade-off between the bias and variance. Therefore, we have to choose λ appropriately by an MS method [16, 19, 18].

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

$$X_{i,j} = \varphi_j(\boldsymbol{x}_i). \tag{2.15}$$

Let $\boldsymbol{D}$ be the diagonal matrix with the i-th diagonal element being the importance weight of the i-th sample:

$$D_{i,i} = \frac{q(\boldsymbol{x}_i)}{p(\boldsymbol{x}_i)}. \tag{2.16}$$

Then the AIWLS estimator $\widehat{\boldsymbol{\alpha}}$ is analytically given by

$$\widehat{\boldsymbol{\alpha}} = \boldsymbol{L}\boldsymbol{y}, \tag{2.17}$$

where

$$\boldsymbol{L} = (\boldsymbol{X}^\top \boldsymbol{D}^\lambda \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{D}^\lambda, \tag{2.18}$$

$$\boldsymbol{y} = (y_1, y_2, \ldots, y_n)^\top. \tag{2.19}$$

2.3 Active Learning (AL) AL is the problem of optimizing the training input density $p(\boldsymbol{x})$ so that the generalization error is minimized[2]:

$$\min_p G(p). \tag{2.20}$$

In order to perform AL, the inaccessible generalization error G has to be estimated. Note that in batch AL, we have to estimate G *before* training samples is observed (cf. MS in Section 2.4). Since our model (2.2) could be misspecified (see Section 2.1), we can not reliably use the traditional OLS-based AL method [5, 3, 6].

Recently, a novel generalization error estimator $\widehat{G}^{(AL)}$ for AL has been developed, which is shown to be reliable even if the model (2.2) is not correctly specified [17]:

$$\widehat{G}^{(AL)} = \mathrm{tr}(\boldsymbol{U}\boldsymbol{L}\boldsymbol{L}^\top), \tag{2.21}$$

where $\boldsymbol{U}$ is the b-dimensional square matrix with the (i, j)-th element

$$U_{i,j} = \int_{\mathcal{D}} \varphi_i(\boldsymbol{x}) \varphi_j(\boldsymbol{x}) q(\boldsymbol{x}) d\boldsymbol{x}. \tag{2.22}$$

Note that $\sigma^2 \widehat{G}^{(AL)}$ corresponds to the variance term V (see Eq.(2.12)). When the model is approximately correct (i.e., the model error δ asymptotically vanishes) and $\lambda \to 1$ (i.e., $\widehat{\boldsymbol{\alpha}}$ is asymptotically unbiased), $\widehat{G}^{(AL)}$ is shown to be a consistent estimator of the expected generalization error (up to the model error δ^2):

$$\sigma^2 \widehat{G}^{(AL)} = \mathop{\mathbb{E}}_{\boldsymbol{\epsilon}} G - \delta^2 + o_p(n^{-1}), \tag{2.23}$$

where $o_p(\cdot)$ denotes the asymptotic order in probability. A sketch of its proof is reviewed in Appendix A. This justifies the use of Eq.(2.21) as an AL criterion.

2.4 Model Selection (MS) MS is the problem of optimizing a model M so that the generalization error is minimized:

$$\min_M G(M). \tag{2.24}$$

In the current setting, the model M refers to the number and type of the basis functions $\{\varphi_i(\boldsymbol{x})\}_{i=1}^b$; the tuning parameter λ in Eq.(2.14) is also included. In order to perform MS, the inaccessible generalization error G has to be estimated. Note that in the MS problem, it is usually assumed that the training samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ have already been observed [1, 14, 15, 4, 16, 19]. Thus

[1] We may further add a regularizer to AIWLS [8]. But for simplicity, we focus on AIWLS without regularizers.

[2] Precisely, this is the *batch* active learning problem where all training input points are designed at once in the beginning.

$\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ are used for estimating the generalization error.

The current situation contains a shift in input distributions, i.e. $p(\boldsymbol{x}) \neq q(\boldsymbol{x})$. Therefore, standard MS methods such as Akaike's information criterion [1] and cross-validation [4] have strong bias and thus are not reliable [22, 19, 18]. Recently, a novel generalization error estimator for MS has been proposed, which posses proper unbiasedness even under the covariate shift [19]:

$$\begin{aligned}\widehat{G}^{(MS)} =& \langle \boldsymbol{ULy}, \boldsymbol{Ly}\rangle - 2\langle \boldsymbol{ULy}, \boldsymbol{L}_1\boldsymbol{y}\rangle \\ & + 2\widehat{\sigma^2}\mathrm{tr}(\boldsymbol{ULL}_1^\top),\end{aligned} \tag{2.25}$$

where

$$\widehat{\sigma^2} = \frac{\|\boldsymbol{y} - \boldsymbol{XL}_0\boldsymbol{y}\|^2}{n-b}. \tag{2.26}$$

$\boldsymbol{L}_0$ and $\boldsymbol{L}_1$ denote $\boldsymbol{L}$ computed with $\lambda = 0$ and $\lambda = 1$, respectively. $\widehat{G}^{(MS)}$ is shown to satisfy

$$\mathop{\mathbb{E}}_{\boldsymbol{\epsilon}} \widehat{G}^{(MS)} = \mathop{\mathbb{E}}_{\boldsymbol{\epsilon}} G - C + \mathcal{O}_p(\delta n^{-\frac{1}{2}}), \tag{2.27}$$

where

$$C = \int_{\mathcal{D}} f(\boldsymbol{x})^2 q(\boldsymbol{x}) d\boldsymbol{x}. \tag{2.28}$$

A sketch of its proof is reviewed in Appendix B. This means that, $\widehat{G}^{(MS)}$ is an exact unbiased estimator of the expected generalization error (up to the constant C) if the model is correctly specified (i.e., the model error δ is zero); for misspecified models, it is an asymptotic unbiased estimator in general, where the asymptotic error is proportional to the model error δ. This justifies the use of Eq.(2.25) as a MS criterion.

2.5 Active Learning with Model Selection (ALMS) The problems of AL and MS share a common goal—minimizing the generalization error (see Eqs.(2.20) and (2.24)). However, they have been studied separately as two independent problems so far. If AL and MS are performed at the same time, the generalization performance would be further improved. We call the problem of simultaneously optimizing training input points and models *active learning with model selection* (ALMS):

$$\min_{p,M} G(p, M). \tag{2.29}$$

This is the problem we would like to solve in this paper.

3 Existing approaches to ALMS

In this section, we discuss strengths and limitations of existing approaches to solving the ALMS problem (2.29).

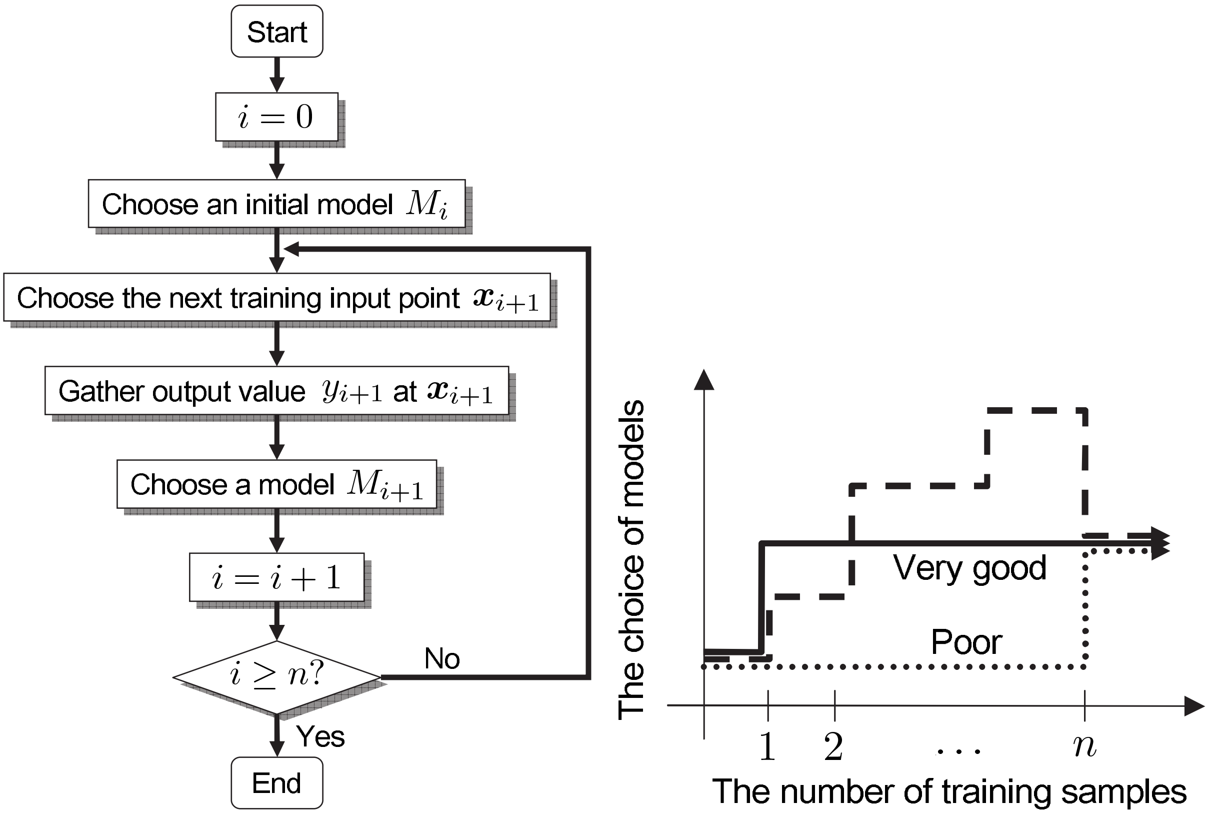
3.1 Direct approach and the AL/MS dilemma

A naive and direct solution to the ALMS problem would be to simultaneously optimize $p(\boldsymbol{x})$ and M. However, this direct approach may not be possible by simply combining existing AL methods and MS methods in a batch manner due to the *AL/MS dilemma*: when selecting the training input density $p(\boldsymbol{x})$ with existing AL methods, the model M must have been fixed [5, 11, 3, 6, 21, 9, 17]. On the other hand, when choosing the model M with existing MS methods, the training input points $\{\boldsymbol{x}_i\}_{i=1}^n$ (or the training input density $p(\boldsymbol{x})$) must have been fixed and the corresponding training output values $\{y_i\}_{i=1}^n$ must have been gathered [1, 14, 15, 4, 16, 19]. For example, the AL criterion (2.21) can not be computed without fixing the model M and the MS criterion (2.25) can not be computed without fixing the training input density $p(\boldsymbol{x})$.

If training input points that are optimal for all model candidates exist, it is possible to perform AL and MS at the same time without regard to the AL/MS dilemma: choose the training input points $\{\boldsymbol{x}_i\}_{i=1}^n$ for some model M by a standard AL method (e.g., Eq.(2.21)), gather corresponding output values $\{y_i\}_{i=1}^n$, and perform MS using a standard method (e.g., Eq.(2.25)). It is shown that such common optimal training input points exist for a class of correctly specified trigonometric polynomial regression models [20]. However, such common optimal training input points may not exist in general and thus the range of application of this approach is limited.

3.2 Sequential approach A standard approach to coping with the above AL/MS dilemma for arbitrary models would be the *sequential approach* [10], i.e., in an iterative manner, a model is chosen by an MS method and the next input point (or a small portion) is optimized for the chosen model by an AL method (see Figure 3(a)).

In the sequential approach, the chosen model M_i varies through the online learning process (see the dashed line in Figure 3(b)). We refer to this phenomenon as the *model drift*. The model drift phenomenon could be a weakness of the sequential approach since the location of optimal training input points depends on the target model in AL; thus a good training input point for one model could be poor for another model (see Section 5.1 for illustrative examples). Depending on the transition of the chosen models, the sequential approach can work very well. For example, when the transition of the model is the solid line in Figure 3(b), most of the training input points are chosen for the finally selected model M_n and the sequential approach has an excellent performance. However, when


Start
$i = 0$
Choose an initial model M_i
Choose the next training input point $\boldsymbol{x}_{i+1}$
Gather output value y_{i+1} at $\boldsymbol{x}_{i+1}$
Choose a model M_{i+1}
$i = i + 1$
$i \geq n?$
No
Yes
End
The choice of models
Very good
Poor
1
2
$\cdots$
n
The number of training samples

(a) Diagram

(b) Transition of chosen models

Figure 3: Sequential approach.

the transition of the model is the dotted line in Figure 3(b), the performance becomes poor since most of the training input points are chosen for other models. Note that we can *not* control the transition of the model properly since we do not know a priori which model will be chosen in the end. Therefore, the performance of the sequential approach is unstable.

Another issue that needs to be taken into account in the sequential approach is that the training input points are not independent and identically distributed (*i.i.d.*) in general—the choice of the $(i+1)$-th training input point $\boldsymbol{x}_{i+1}$ depends on the previously gathered samples $\{(\boldsymbol{x}_j, y_j)\}_{j=1}^{i}$. Since standard AL and MS methods require the i.i.d. assumption for establishing their statistical properties such as consistency or unbiasedness, they may not be directly employed in the sequential approach [2]. The AL criterion (2.21) and MS criterion (2.25) also suffer from the violation of the i.i.d. condition, and loose their consistency and unbiasedness. However, this problem can be settled by slightly modifying the criteria, which is an advantage of the AL criterion (2.21) and MS criterion (2.25): Suppose we draw u input points from $p^{(i)}(\boldsymbol{x})$ in each iteration (let $n = uv$, where v is the number of iterations). If u tends to infinity, simply redefining the diagonal matrix D as follows makes $\widehat{G}^{(AL)}$ and $\widehat{G}^{(MS)}$ still consistent and asymptotically unbiased:

$$D_{k,k} = \frac{q(\boldsymbol{x}_k)}{p^{(i)}(\boldsymbol{x}_k)}, \tag{3.30}$$

where

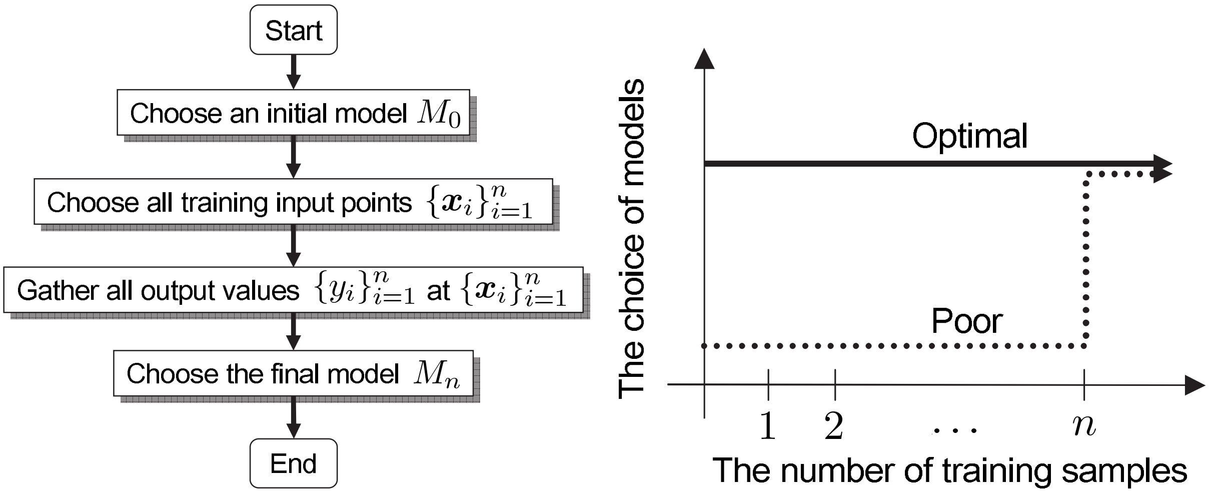
$$k = (i-1)u + j, \tag{3.31}$$

$$i = 1, 2, \ldots, v, \tag{3.32}$$

$$j = 1, 2, \ldots, u. \tag{3.33}$$

4 Proposed approach: ensemble active learning (EAL)

In the previous section, we pointed out potential limitations of existing approaches. In this section, we propose a new ALMS method that can cope with the above limitations.


Start
Choose an initial model M_0
Choose all training input points $\{\boldsymbol{x}_i\}_{i=1}^n$
Gather all output values $\{y_i\}_{i=1}^n$ at $\{\boldsymbol{x}_i\}_{i=1}^n$
Choose the final model M_n
End
The choice of models
Optimal
Poor
1
2
. . .
n
The number of training samples

(a) Diagram

(b) Transition of chosen models

Figure 4: Batch approach.

4.1 Batch approach An alternative approach to ALMS is to choose all the training input points for an initially chosen model M_0. We refer to this approach as the *batch approach* (see Figure 4(a)). Due to the batch nature, this approach does not suffer from the model drift (cf. Figure 3(b)); the batch approach can be optimal in terms of AL if an initially chosen model M_0 agrees with the finally chosen model M_n (see the solid line in Figure 4(b)).

In order to choose the initial model M_0, we may need a generalization error estimator that can be computed before observing training samples—for example, the generalization error estimator (2.21). However, this does not work well since Eq.(2.21) only evaluates the variance of the estimator (see Eq.(2.12)); thus using Eq.(2.21) for choosing the initial model M_0 simply results in always selecting the simplest model from the candidates. Note that this problem is not specific to the generalization error estimator (2.21), but is common to most generalization error estimators since it is generally not possible to estimate the bias of the estimator (see Eq.(2.11)) before observing training samples. Therefore, in practice, we may have to choose the initial model M_0 *randomly.* If we have some prior preference of models, $P(M)$, we may draw the initial model according to it; otherwise we just have to choose the initial model M_0 randomly from the uniform distribution.

Due to the randomness of the initial model choice, the performance of the batch approach may be unstable (see the dashed line in Figure 4(b)).

4.2 Ensemble active learning (EAL) The weakness of the batch approach lies in the fact that the training input points chosen by an AL method are *overfitted* to the initially chosen model—the training input points optimized for the initial model could be poor if a different model is chosen later.

We may reduce the risk of overfitting by not optimizing the training input density $p(\boldsymbol{x})$ *specifically* for a single model, but by optimizing it for *all* model candidates (see Figure 5). This allows all the models to contribute to the optimization of the training input density and thus we can hedge the risk of overfitting to a single (possibly inferior) model. Since this approach could be viewed as applying a popular idea of *ensemble learning* to the problem of AL, we refer to the proposed approach as *ensemble active learning* (EAL).

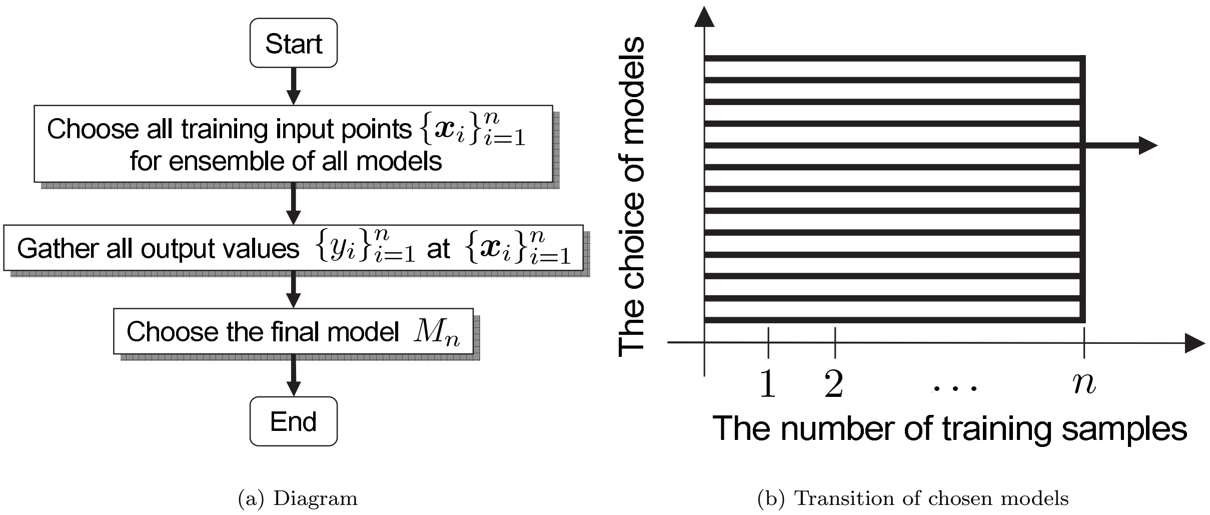
This idea could be realized by determining the training input density $p(\boldsymbol{x})$ so that the *expected* generalization error over *all* model candidates is minimized:

$$\min_{p} \widehat{G}^{(EAL)}(p), \tag{4.34}$$

where

$$\widehat{G}^{(EAL)}(p) = \sum_{M} \widehat{G}^{(AL)}_{M}(p)P(M). \tag{4.35}$$

$\widehat{G}^{(AL)}_{M}$ is the value of $\widehat{G}^{(AL)}$ for a model M and $P(M)$ is the prior preference of the model M. If no prior information on goodness of the models is available, the uniform prior may be simply used. In Section 5, we experimentally show that this ensemble approach significantly outperforms the sequential and batch approaches.


Start
Choose all training input points $\{\boldsymbol{x}_i\}_{i=1}^n$ for ensemble of all models
Gather all output values $\{y_i\}_{i=1}^n$ at $\{\boldsymbol{x}_i\}_{i=1}^n$
Choose the final model M_n
End
The choice of models
1
2
...
n
The number of training samples

(a) Diagram

(b) Transition of chosen models

Figure 5: The proposed ensemble approach.

We can also consider a variant of the ensemble approach where the magnitude of each AL criterion $\widehat{G}^{(AL)}(p)$ is normalized:

$$\min_p \widehat{G}^{(nEAL)}(p), \tag{4.36}$$

where

$$\widehat{G}^{(nEAL)}(p) = \sum_M \frac{\widehat{G}_M^{(AL)}(p)}{\sum_{p'} \widehat{G}_M^{(AL)}(p')} P(M). \tag{4.37}$$

This variant may have an effect of reducing the influence of poor models. However, our preliminary experiments showed that the use of the normalization scheme does not make a big difference. For this reason, we do not go into the detail any further.

It is also possible to implement the ensemble idea in AL by allowing each model M to choose n_M training input points, where

$$n_M \propto P(M) \tag{4.38}$$

and

$$\sum_M n_M = n. \tag{4.39}$$

More specifically, we let each model M select the training input density $p_M(\boldsymbol{x})$ by

$$p_M(\boldsymbol{x}) = \operatorname*{argmin}_p \widehat{G}_M^{(AL)}(p), \tag{4.40}$$

where $\widehat{G}_M^{(AL)}(p)$ is computed based on n_M training input points (see Eq.(2.21)). Then each model M draws n_M training input points from $p_M(\boldsymbol{x})$. However, our preliminary experiments showed that this alternative idea does not work well. For this reason, we omit the detail.

5 Numerical experiments

In this section, we quantitatively compare the proposed EAL method with other approaches through numerical experiments.

5.1 Toy data set Here we illustrate how the ensemble (Section 4.2), batch (Section 4.1), and sequential (Section 3.2) methods behave using a toy one-dimensional data set.

Let the input dimension be $d = 1$ and the target function $f(x)$ be the following third order polynomial (see the top graph of Figure 6):

$$f(x) = 1 - x + x^2 + r(x), \tag{5.41}$$

where, for $\delta = 0.05$,

$$r(x) = \delta \frac{z^3 - 3z}{\sqrt{6}} \quad \text{with} \quad z = \frac{x - 0.2}{0.4}. \tag{5.42}$$

Let the test input density $q(x)$ be the Gaussian density with mean 0.2 and standard deviation 0.4, which is assumed to be known in this illustrative simulation. We choose the training input density $p(x)$ from a set of Gaussian densities with mean 0.2 and standard deviation $0.4c$, where

$$c = 0.8, 0.9, 1.0, \ldots, 2.5. \tag{5.43}$$

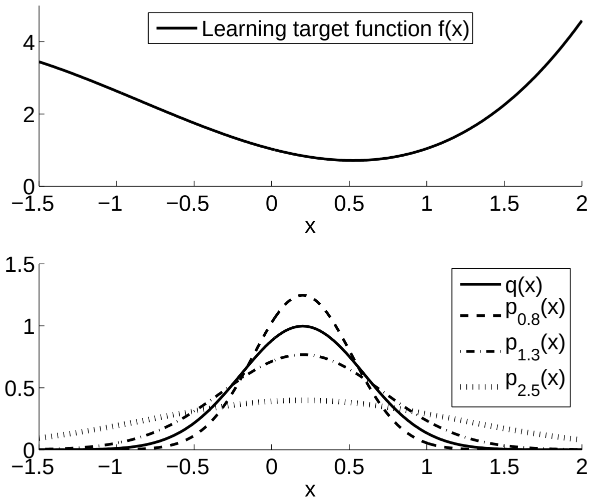


Figure 6: Target function, training input density $p_c(x)$, and test input density $q(x)$.

These density functions are illustrated in the bottom graph of Figure 6. We add i.i.d. Gaussian noise with mean zero and standard deviation 0.3 to the training output values.

Let the number of basis functions be $b = 3$ and the basis functions be

$$(5.44) \qquad \varphi_i(x) = x^{i-1} \text{ for } i = 1, 2, \ldots, b.$$

Note that the target function $f(x)$ is not realizable since the model is the second order polynomial. Here, for illustration purposes, we use the above fixed basis functions[3] and focus on choosing λ by MS; λ is selected from

$$(5.45) \qquad \lambda = 0, 0.5, 1.$$

First, we investigate the dependency between the goodness of the training input density (i.e., c) and the model (i.e., λ). For each λ and each c, we draw training input points $\{x_i\}_{i=1}^{100}$ and gather output values $\{y_i\}_{i=1}^{100}$. Then we learn the parameter $\widehat{\boldsymbol{\alpha}}$ by AIWLS and compute the generalization error G. The mean G over 1000 trials as a function of c for each λ is depicted in Figure 7(a). This graph underlines that the best training input density c could strongly depend on the model λ, implying that a training input density that is good for one model could be poor for others. For example, when the training input density is optimized for the model $\lambda = 0$, $c = 1.1$ would be an excellent choice. However, $c = 1.1$ is not so suitable for other models $\lambda = 0.5, 1$. This figure illustrate a possible weakness of the batch method: when an initially chosen model is significantly different from the finally chosen model, the training input points optimized for the initial model could be less useful for the final model and the performance is degraded.

Next, we investigate the behavior of the sequential approach. In our implementation, 10 training input points are chosen at each iteration. Figure 7(b) depicts the transition of the frequency of chosen λ in the sequential learning process over 1000 trials. It shows that the choice of models varies over the learning process; a smaller λ (which has smaller variance thus low complexity) is favored in the beginning, but a larger λ (which has larger variance thus higher complexity) tends to be chosen as the number of training samples increases. Figure 7 illustrates a possible weakness of the sequential method: the target model drifts during the sequential learning process (from small λ to large λ) and the training input points designed in an early stage (for $\lambda = 0$) could be poor for the finally chosen model ($\lambda = 1$).

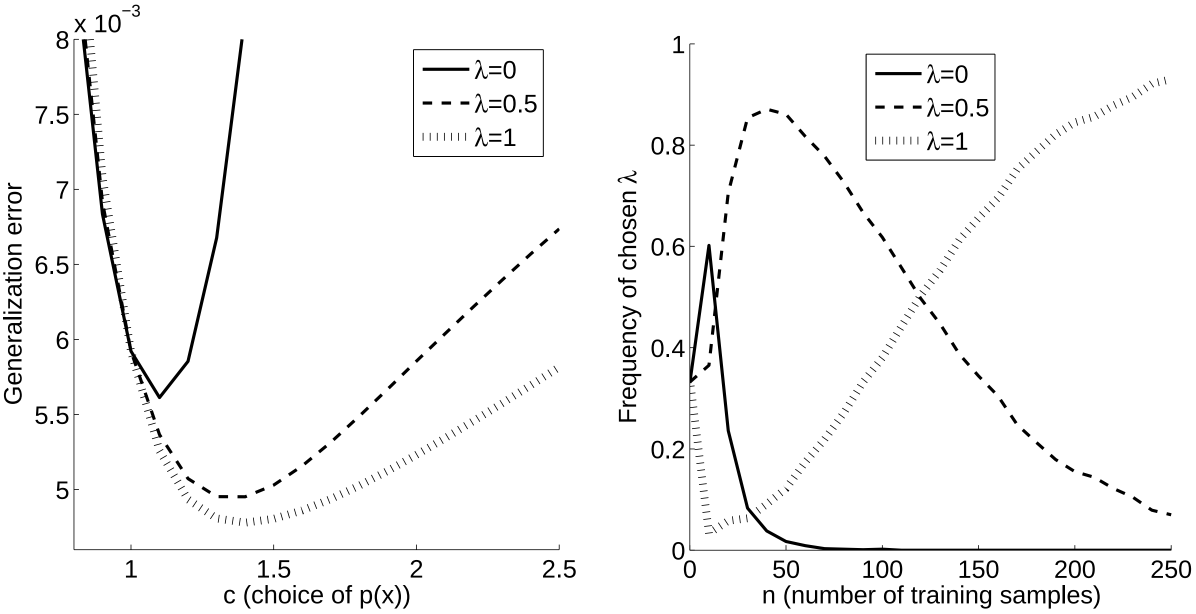
Finally, we investigate the generalization performance of each method when the number of training samples to gather is

$$(5.46) \qquad n = 100, 150, 200, 250.$$

Table 1 describes the means and standard deviations of the generalization error obtained by the sequential, batch, and ensemble methods; as a baseline, we also included the result of passive learning, i.e., the training input points $\{x_i\}_{i=1}^{n}$ are drawn from the test input density $q(x)$ (or equivalently $c = 1$). The table shows that all three ALMS methods tend to outperform passive learning. However, the improvement of the sequential method is not so significant, which would be caused by the model drift phenomenon (see Figure 7). The batch method also does not provide significant improvement due to the overfitting to the randomly chosen initial model (see Figure 7(a)). On the other hand, the proposed ensemble method does not suffer from these problems and works significantly better than other methods—the best method and comparable ones by the *t-test* at the significance level 5% [7] are marked by '○' in the table.

5.2 Benchmark data sets Here we evaluate whether the advantages of the proposed method are still valid under more realistic settings. For this purpose, we use eight regression benchmark data sets provided by DELVE [13]: *Bank-8fm, Bank-8fh, Bank-8nm, Bank-8nh, Pumadyn-8fm, Pumadyn-8fh, Pumadyn-8nm*, and *Pumadyn-8nh*. Each data set includes 8192 samples, consisting of 8-dimensional input points and 1-

[3] Note that we can also choose the order of polynomials by MS.



(a)

(b)

Figure 7: (a) The mean generalization error over 1000 trials as a function of training input density c for each λ (when $n = 100$). (b) Frequency of chosen λ over 1000 trials as a function of the number of training samples.

dimensional output values. For convenience, every attribute is normalized into $[0, 1]$.

Suppose we are given all 8192 *input* points (i.e., unlabeled samples). Note that output values are kept unknown at this point. From this pool of unlabeled samples, we choose $n = 200$ input points $\{\boldsymbol{x}_i\}_{i=1}^{n}$ for training and observe the corresponding output values $\{y_i\}_{i=1}^{n}$. The task is to predict the output values of all 8192 samples.

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

$$(5.47) \quad q(\boldsymbol{x}) = \frac{1}{(2\pi\widehat{\gamma}_{MLE}^2)^{\frac{d}{2}}} \exp\left(-\frac{\|\boldsymbol{x} - \widehat{\boldsymbol{\mu}}_{MLE}\|^2}{2\widehat{\gamma}_{MLE}^2}\right),$$

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

$$(5.48) \qquad c = 0.5, 0.6, 0.7, \ldots, 1.5.$$

Let the number of basis functions be $b = 100$ and let the basis functions be Gaussian functions with center $\boldsymbol{t}_i$ and width h: for $i = 1, 2, \ldots, b$,

$$(5.49) \qquad \varphi_i(\boldsymbol{x}) = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{t}_i\|^2}{2h^2}\right).$$

The centers $\{\boldsymbol{t}_i\}_{i=1}^{b}$ are randomly chosen from the pool of unlabeled samples. In this experiment, we fix the number of basis functions at $b = 100$ and choose λ from

$$(5.50) \qquad \lambda = 0, 0.5, 1,$$

and standard deviation h of Gaussian basis functions from

$$(5.51) \qquad h = 0.4, 0.8, 1.2.$$

We again compare the proposed ensemble method with the batch, sequential, and passive methods. In this simulation, we can not create the training input points in an arbitrary location because we only have 8192 samples in the pool. Here, we first create provisional input points following the determined training input density, and then choose the input points from the pool of unlabeled samples that are closest to the provisional input points. In this simulation, the expectation over the test input density $q(\boldsymbol{x})$ in the matrix $\boldsymbol{U}$ (see Eq.(2.22)) is calculated by the empirical average over

Table 1: Means and standard deviations of generalization error for the toy data set. All values in the table are multiplied by 10^3. The best method in terms of the mean generalization error and comparable methods according to the t-test at the significance level 5% are marked by '∘'.

n	Passive	Sequential	Batch	Ensemble
100	5.92±3.28	5.57±2.75	5.65±2.92	$^{\circ}$5.12±2.50
150	4.77±2.18	4.43±1.77	4.64±1.91	$^{\circ}$4.11±1.55
200	4.21±1.75	3.89±1.40	4.19±1.60	$^{\circ}$3.68±1.19
250	3.78±1.32	3.47±1.02	3.91±1.42	$^{\circ}$3.35±0.92

all 8192 unlabeled samples since the true test error is also calculated as such. For each data set, we run this simulation 1000 times, by changing the template points $\{\boldsymbol{t}_i\}_{i=1}^{b}$ in each run (thus the choice of training input points is also changed in each trial).

Table 2 describes the mean and standard deviation of the generalization error by each method. All the values are normalized by the mean generalization error of the passive learning method for better comparison. In the table, the best method in terms of the mean generalization error and comparable methods according to the *t-test* at the significance level 5% [7] are marked by '∘' This shows that all three ALMS methods perform better than the passive learning method. Among them, the proposed ensemble method tends to work significantly better than other methods.

Based on the simulation results, we conclude that the proposed ensemble approach is useful in ALMS scenarios; thus it could be a promising alternative to the *de facto* standard sequential approach.

6 Conclusions

So far, the problems of active learning (AL) and model selection (MS) have been studied as two independent problems, although they both share a common goal of minimizing the generalization error. We suggested that by simultaneously performing AL and MS—which we called *active learning with model selection* (ALMS), a better generalization capability could be achieved. We pointed out that the sequential approach, which would be a common approach to ALMS, can perform poorly due to the model drift phenomenon (Section 3.2). To overcome the limitations of the sequential approach, we proposed a new approach called *ensemble active learning* (EAL), which performs AL not only for a single model, but for an ensemble of models (Section 4.2). We have demonstrated through simulations that the proposed EAL method compares favorably with other approaches (Section 5).

In theory, IWLS is asymptotically unbiased as long as the support of training and test input distributions are equivalent. However, in practical situations with finite samples, IWLS may not work properly if these two distributions are less overlapped. It is important to theoretically investigate how robust IWLS is when training and test input distributions are significantly different.

In real applications, we are often given unlabeled samples and want to choose the best samples to label. Such a situation is referred to as *pool-based* scenarios. In our experiments, we estimated the input density from the unlabeled samples and showed that the proposed approach significantly outperforms passive learning. However, it would be more promising to develop a method that can directly deal with a finite number of unlabeled samples.

Although we focused on regression problems in this paper, the idea of EAL is applicable in any supervised learning scenarios, given that a suitable *batch* AL method is available. This implies that, in principle, it is possible to extend the proposed EAL method to classification problems. However, to the best of our knowledge, there is no reliable batch AL method in classification tasks. Therefore, developing an ALMS method for classification is still a challenging open problem, which needs to be investigated.

A Proof of Eq.(2.23)

First, we show the consistency (2.23) when $\lambda = 1$. A simple calculation yields that the bias B and the variance V can be expressed as

$$B = \langle \boldsymbol{U}(\underset{\boldsymbol{\epsilon}}{\mathbb{E}}\,\widehat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}^*), \underset{\boldsymbol{\epsilon}}{\mathbb{E}}\,\widehat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}^*\rangle, \tag{1.52}$$

$$V = \underset{\boldsymbol{\epsilon}}{\mathbb{E}}\langle \boldsymbol{U}(\widehat{\boldsymbol{\alpha}} - \underset{\boldsymbol{\epsilon}}{\mathbb{E}}\,\widehat{\boldsymbol{\alpha}}), \widehat{\boldsymbol{\alpha}} - \underset{\boldsymbol{\epsilon}}{\mathbb{E}}\,\widehat{\boldsymbol{\alpha}}\rangle. \tag{1.53}$$

Let

$$\boldsymbol{z}_g = (g(\boldsymbol{x}_1), g(\boldsymbol{x}_2), \ldots g(\boldsymbol{x}_n))^\top, \tag{1.54}$$

$$\boldsymbol{z}_r = (r(\boldsymbol{x}_1), r(\boldsymbol{x}_2), \ldots r(\boldsymbol{x}_n))^\top. \tag{1.55}$$

By definition, it holds that

$$\boldsymbol{z}_g = \boldsymbol{X}\boldsymbol{\alpha}^*. \tag{1.56}$$

Table 2: Means and standard deviations of the generalization error for the DELVE data sets. All values are normalized by the mean generalization error of the passive learning method. The best method in terms of the mean normalized generalization error and comparable methods according to the t-test at the significance level 5% are marked by '∘'.

Data set	Passive	Sequential	Batch	Ensemble
bank-8fm	1.00±1.22	0.59±0.85	∘0.46±0.25	∘0.45±0.28
bank-8fh	1.00±0.42	0.53±0.22	0.46±0.18	∘0.44±0.11
bank-8nm	1.00±0.76	0.63±0.19	0.58±0.21	∘0.56±0.10
bank-8nh	1.00±0.28	0.61±0.19	0.53±0.14	∘0.51±0.11
pumadyn-8fm	1.00±0.22	∘0.83±0.36	0.92±0.68	0.91±0.73
pumadyn-8fh	1.00±0.17	0.80±0.17	0.76±0.22	∘0.71±0.19
pumadyn-8nm	1.00±0.18	0.86±0.15	0.85±0.20	∘0.81±0.18
pumadyn-8nh	1.00±0.19	0.85±0.14	0.81±0.17	∘0.77±0.15

Then we have

$$\begin{aligned}\mathop{\mathbb{E}}_{\boldsymbol{\epsilon}}\widehat{\boldsymbol{\alpha}}-\boldsymbol{\alpha}^* =&\boldsymbol{L}(\boldsymbol{z}_g+\delta\boldsymbol{z}_r)-\boldsymbol{\alpha}^*\\ =&(\tfrac{1}{n}\boldsymbol{X}^\top\boldsymbol{D}\boldsymbol{X})^{-1}\tfrac{1}{n}\boldsymbol{X}^\top\boldsymbol{D}(\boldsymbol{X}\boldsymbol{\alpha}^*+\delta\boldsymbol{z}_r)\\ &-\boldsymbol{\alpha}^*\\ =&\delta(\tfrac{1}{n}\boldsymbol{X}^\top\boldsymbol{D}\boldsymbol{X})^{-1}\tfrac{1}{n}\boldsymbol{X}^\top\boldsymbol{D}\boldsymbol{z}_r.\end{aligned}\tag{1.57}$$

By the law of large numbers [12], we have

$$\begin{aligned}&\lim\nolimits_{n\to\infty}[\tfrac{1}{n}\boldsymbol{X}^\top\boldsymbol{D}\boldsymbol{X}]_{i,j}\\ &=\lim_{n\to\infty}\left(\frac{1}{n}\sum_{k=1}^{n}\frac{q(\boldsymbol{x}_k)}{p(\boldsymbol{x}_k)}\varphi_i(\boldsymbol{x}_k)\varphi_j(\boldsymbol{x}_k)\right)\\ &=\int_{\mathcal{D}}\frac{q(\boldsymbol{x})}{p(\boldsymbol{x})}\varphi_i(\boldsymbol{x})\varphi_j(\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x}\\ &=\mathcal{O}_p(1).\end{aligned}\tag{1.58}$$

Furthermore, by the central limit theorem [12], it holds for sufficiently large n,

$$\begin{aligned}[\tfrac{1}{n}\boldsymbol{X}^\top\boldsymbol{D}\boldsymbol{z}_r]_i&=\frac{1}{n}\sum_{k=1}^{n}r(\boldsymbol{x}_k)\varphi_i(\boldsymbol{x}_k)\frac{q(\boldsymbol{x}_k)}{p(\boldsymbol{x}_k)}\\ &=\int_{\mathcal{D}}r(\boldsymbol{x})\varphi_i(\boldsymbol{x})\frac{q(\boldsymbol{x})}{p(\boldsymbol{x})}p(\boldsymbol{x})d\boldsymbol{x}+\mathcal{O}_p(n^{-\frac{1}{2}})\\ &=\mathcal{O}_p(n^{-\frac{1}{2}}),\end{aligned}\tag{1.59}$$

where the last equality follows from Eq.(2.8). Given

$$\boldsymbol{U}=\mathcal{O}_p(1),\tag{1.60}$$

we have

$$B=\mathcal{O}_p(\delta^2 n^{-1}).\tag{1.61}$$

Since

$$\begin{aligned}\boldsymbol{L}\boldsymbol{L}^\top&=(\tfrac{1}{n}\boldsymbol{X}^\top\boldsymbol{D}\boldsymbol{X})^{-1}\tfrac{1}{n^2}\boldsymbol{X}^\top\boldsymbol{D}^2\boldsymbol{X}(\tfrac{1}{n}\boldsymbol{X}^\top\boldsymbol{D}\boldsymbol{X})^{-1}\\ &=\mathcal{O}_p(n^{-1}),\end{aligned}\tag{1.62}$$

we have

$$\begin{aligned}V&=\sigma^2\mathrm{tr}(\boldsymbol{U}\boldsymbol{L}\boldsymbol{L}^\top)\\ &=\mathcal{O}_p(n^{-1}).\end{aligned}\tag{1.63}$$

Thus, if $\delta=o_p(1)$,

$$\begin{aligned}\mathop{\mathbb{E}}_{\boldsymbol{\epsilon}}G&=B+V+\delta^2\\ &=o_p(n^{-1})+\sigma^2\widehat{G}^{(AL)}+\delta^2,\end{aligned}\tag{1.64}$$

which results in Eq.(2.23). We may establish the same argument if λ is asymptotically one.

B Proof of Eq.(2.27)

We show the asymptotic unbiasedness (2.27). A simple calculation yields that the generalization error G is expressed as

$$G=\langle\boldsymbol{U}\widehat{\boldsymbol{\alpha}},\widehat{\boldsymbol{\alpha}}\rangle-2\langle\boldsymbol{U}\widehat{\boldsymbol{\alpha}},\boldsymbol{\alpha}^*\rangle+C.\tag{2.65}$$

Since

$$\mathop{\mathbb{E}}_{\boldsymbol{\epsilon}}\langle\boldsymbol{U}\widehat{\boldsymbol{\alpha}},\boldsymbol{\alpha}^*\rangle=\langle\boldsymbol{U}\boldsymbol{L}(\boldsymbol{z}_g+\delta\boldsymbol{z}_r),\boldsymbol{L}_1\boldsymbol{z}_g\rangle,\tag{2.66}$$

$$\begin{aligned}\mathop{\mathbb{E}}_{\boldsymbol{\epsilon}}\langle\boldsymbol{U}\widehat{\boldsymbol{\alpha}},\boldsymbol{L}_1\boldsymbol{y}\rangle=&\langle\boldsymbol{U}\boldsymbol{L}(\boldsymbol{z}_g+\delta\boldsymbol{z}_r),\boldsymbol{L}_1(\boldsymbol{z}_g+\delta\boldsymbol{z}_r)\rangle\\ &+\sigma^2\mathrm{tr}(\boldsymbol{U}\boldsymbol{L}\boldsymbol{L}_1^\top),\end{aligned}\tag{2.67}$$

we have

$$\begin{aligned}\mathop{\mathbb{E}}_{\boldsymbol{\epsilon}}G-C-\mathop{\mathbb{E}}_{\boldsymbol{\epsilon}}\widehat{G}^{(MS)}=&2\langle\boldsymbol{U}\boldsymbol{L}(\boldsymbol{z}_g+\delta\boldsymbol{z}_r),\delta\boldsymbol{L}_1\boldsymbol{z}_r\rangle\\ &+2(\mathop{\mathbb{E}}_{\boldsymbol{\epsilon}}\widehat{\sigma^2}-\sigma^2)\mathrm{tr}(\boldsymbol{U}\boldsymbol{L}\boldsymbol{L}_1^\top).\end{aligned}\tag{2.68}$$

Eqs.(1.58) and (1.59) imply

$$\boldsymbol{L}_1 \boldsymbol{z}_r = \mathcal{O}_p(n^{-\frac{1}{2}}). \tag{2.69}$$

Thus the first term in the right-hand side of Eq.(2.68) is of $\mathcal{O}_p(\delta n^{-\frac{1}{2}})$. Since

$$\mathrm{tr}(\boldsymbol{U}\boldsymbol{L}\boldsymbol{L}_1^\top) = \mathcal{O}_p(n^{-1}), \tag{2.70}$$

$$\mathop{\mathbb{E}}_{\boldsymbol{\epsilon}} \widehat{\sigma^2} = \sigma^2 + \frac{\delta^2 \|\boldsymbol{G}\boldsymbol{z}_r\|^2}{\mathrm{tr}(\boldsymbol{G})}, \tag{2.71}$$

where

$$\boldsymbol{G} = \boldsymbol{I} - \boldsymbol{X}\boldsymbol{L}_0, \tag{2.72}$$

and $\boldsymbol{I}$ is the identity matrix, the second term in the right-hand side of Eq.(2.68) is of $\mathcal{O}_p(\delta^2 n^{-1})$. This establishes Eq.(2.27).

References

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

[2] F. Bach. Active learning for misspecified generalized linear models. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*. MIT Press, Cambridge, MA, 2007.

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

[4] P. Craven and G. Wahba. Smoothing noisy data with spline functions: Estimating the correct degree of smoothing by the method of generalized cross-validation. *Numerische Mathematik*, 31:377–403, 1979.

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

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

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

[8] J. Huang, A. Smola, A. Gretton, K. M. Borgwardt, and B. Schölkopf. Correcting sample selection bias by unlabeled data. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 601–608. MIT Press, Cambridge, MA, 2007.

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

[10] D. J. C. MacKay. Bayesian interpolation. *Neural Computation*, 4(3):415–447, 1992.

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

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

[13] C. E. Rasmussen, R. M. Neal, G. E. Hinton, D. van Camp, M. Revow, Z. Ghahramani, R. Kustra, and R. Tibshirani. The DELVE manual, 1996.

[14] J. Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.

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

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

[17] M. Sugiyama. Active learning in approximately linear regression based on conditional expectation of generalization error. *Journal of Machine Learning Research*, 7:141–166, Jan. 2006.

[18] M. Sugiyama, M. Krauledat, and K.-R. Müller. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8:985–1005, May 2007.

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

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

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

[22] B. Zadrozny. Learning and evaluating classifiers under sample selection bias. In *Proceedings of the Twenty-First International Conference on Machine Learning*, New York, NY, 2004. ACM Press.