IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol.E89-A, no.8, pp.2216-2225, 2006.

Analytic Optimization of Shrinkage Parameters based on Regularized Subspace Information Criterion

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

Keisuke Sakurai (sakura@isl.titech.ac.jp)
Department of Computational Intelligence and Systems Science,
Tokyo Institute of Technology,
4259 Nagatsuta-cho, Midori-ku, Yokohama, 226-8502, Japan

Abstract

For obtaining a higher level of generalization capability in supervised learning, model parameters should be optimized, i.e., they should be determined in such a way that the generalization error is minimized. However, since the generalization error is inaccessible in practice, model parameters are usually determined in such a way that an estimate of the generalization error is minimized. A standard procedure for model parameter optimization is to first prepare a finite set of candidates of model parameter values, estimate the generalization error for each candidate, and then choose the best one from the candidates. If the number of candidates is increased in this procedure, the optimization quality may be improved. However, this in turn increases the computational cost. In this paper, we give methods for *analytically* finding the optimal model parameter value from a set of *infinitely* many candidates. This maximally enhances the optimization quality while the computational cost is kept reasonable.

Keywords

supervised learning, generalization capability, model selection, shrinkage estimator, regularized subspace information criterion.

1 Introduction

The goal of supervised learning is to estimate an unknown input-output relation from samples, which is mathematically formulated as a function approximation problem. If the learning target function is accurately learned, the output values for unlearned input points can be estimated. This is called the generalization capability. The level of generalization capability is evaluated by the 'closeness' between the learned function and the true function, i.e., the generalization error. We want to obtain the learned function that minimizes the generalization error. The learned function usually depends on model parameters such as the regularization parameter. Therefore, in order to obtain a better function, the model parameters should be chosen appropriately, i.e., so that the generalization error is minimized.

However, since the true learning target function is unknown, the generalization error can not be directly calculated. A standard approach to coping with this problem is to determine the model parameters so that an estimate of the generalization error is minimized. So far, a large number of generalization error estimators have been proposed [12, 13, 1, 16, 11, 9, 20, 19]. Most of the existing generalization error estimators including all the methods cited above are justified by the unbiasedness in some sense. That is, they are good estimators of the generalization error *on average*. This implies that they could be inaccurate for *single trial* since they may have large variance.

To cope with problem, a *regularized* generalization error estimator has been proposed [18], which is called the regularized subspace information criterion (RSIC). RSIC is no longer unbiased, but has smaller variance so is more reliable than unbiased generalization error estimators. RSIC includes an additional tuning parameter in the generalization error estimator itself. In order to perform model selection well by RSIC, this tuning parameter should be determined appropriately. The paper [18] gave an objective and useful criterion for determining the tuning parameter.

An existing model selection procedure based on RSIC is to search the best model parameter and the best tuning parameter within finite sets of candidates, i.e., a naive grid search. If the numbers of candidates are increased in this procedure, the optimization quality may be improved. However, this in turn increases the computational cost. Therefore, RSIC could be rather demanding in computation time. Note that some greedy optimization strategy such as binary search may also be used instead, but it can get stuck in one of the local optima so may not be reliable.

In this paper, we propose novel methods to alleviate this problem. Our approach is to *analytically* derive the optimal values of the model parameter and/or the tuning parameter from sets of *infinitely* many candidates. This enables us to access to the optimal solution within moderate computation time.

2 Formulation of Supervised Learning

In this section, we formulate the supervised learning problem.

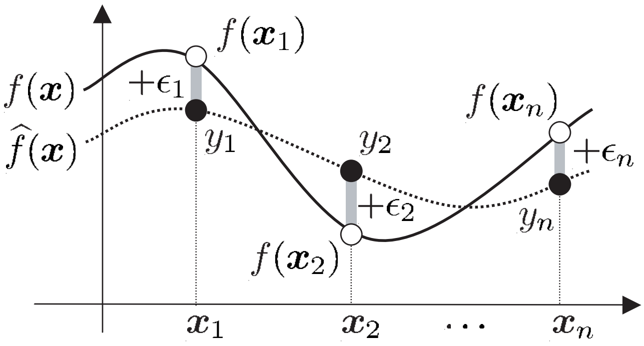


Figure 1: Supervised learning problem.

Let us consider the problem of approximating a function from training samples. Let $f(\boldsymbol{x})$ be the learning target function, which is a real-valued function defined on $\mathcal{D} \subset \mathbb{R}^d$. We assume that $f(\boldsymbol{x})$ belongs to a reproducing kernel Hilbert space $\mathcal{H}$ [3, 24, 23]. Note that $\mathcal{H}$ is generally infinite dimensional. We denote the reproducing kernel of $\mathcal{H}$ by $K(\boldsymbol{x}, \boldsymbol{x}')$. Let $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ be the training samples, where $\boldsymbol{x}_i \in \mathcal{D}$ is an input point and $y_i \in \mathbb{R}$ is an output value. We assume that the output value y_i is degraded by i.i.d. Gaussian noise ϵ_i with mean zero and variance σ^2:

$$y_i = f(\boldsymbol{x}_i) + \epsilon_i. \tag{1}$$

The input points $\{\boldsymbol{x}_i\}_{i=1}^n$ could be either random or deterministic. The above formulation is summarized in Figure 1.

Let $\widehat{f}(\boldsymbol{x})$ be a learned function obtained from training samples. The goal of supervised learning is to obtain the best approximation to the target function. To this end, we need to define the "goodness" measure of $\widehat{f}(\boldsymbol{x})$. In this paper, we measure the goodness of $\widehat{f}(\boldsymbol{x})$ by

$$\|\widehat{f} - f\|^2, \tag{2}$$

where $\|\cdot\|$ is the norm in the reproducing kernel Hilbert space $\mathcal{H}$. Since $\widehat{f}(\boldsymbol{x})$ usually depends on the noise $\{\epsilon_i\}_{i=1}^n$, we consider the expectation of the above goodness measure over the noise. This quantity can be decomposed as

$$\mathbb{E}_\epsilon \|\widehat{f} - f\|^2 = \mathbb{E}_\epsilon \|\widehat{f}\|^2 - 2\mathbb{E}_\epsilon \langle \widehat{f}, f \rangle + \|f\|^2, \tag{3}$$

where $\mathbb{E}_\epsilon$ denotes the expectation over the noise and $\langle \cdot, \cdot \rangle$ denotes the inner product in $\mathcal{H}$. Since the third term $\|f\|^2$ is a constant which does not depend on $\widehat{f}(\boldsymbol{x})$, we ignore it and define the rest by G:

$$G = \mathbb{E}_\epsilon \|\widehat{f}\|^2 - 2\mathbb{E}_\epsilon \langle \widehat{f}, f \rangle. \tag{4}$$

We call G the *generalization error*. In this framework, we do *not* take the expectation of the generalization error over the training input points $\{\boldsymbol{x}_i\}_{i=1}^n$, which is often done in literature [1, 24, 4, 6]. Thus our framework is more *data-dependent* than the others (For the advantages of the data-dependent framework, see the papers [20, 19, 17]).

Now our goal is formalized: we want to learn $\widehat{f}(\boldsymbol{x})$ from training samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$ so that the generalization error G is minimized. To this end, we need to define a search space for $\widehat{f}(\boldsymbol{x})$. The broadest choice would be the function space $\mathcal{H}$ itself, but it is hard to deal with since $\mathcal{H}$ is generally infinite dimensional. To alleviate this problem, we employ the following kernel model for learning [10, 14, 15].

$$\widehat{f}(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i K(\boldsymbol{x}, \boldsymbol{x}_i), \tag{5}$$

where $\{\alpha_i\}_{i=1}^{n}$ are parameters to be learned. Note that this form is known to be a minimizer of some regularized functional in $\mathcal{H}$ [10].

Let

$$\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_n)^{\top}, \tag{6}$$

$$\boldsymbol{y} = (y_1, y_2, \ldots, y_n)^{\top}, \tag{7}$$

where $^{\top}$ denotes the transpose. In this paper, we focus on the cases where the parameter vector $\boldsymbol{\alpha}$ is learned in a linear fashion, i.e., $\boldsymbol{\alpha}$ is obtained by

$$\boldsymbol{\alpha} = \boldsymbol{L}\boldsymbol{y}, \tag{8}$$

where $\boldsymbol{L}$ is an n-dimensional matrix which is independent of the noise. We call $\boldsymbol{L}$ the *learning matrix*.

Consequently, the problem of learning $\widehat{f}(\boldsymbol{x})$ is converted into the problem of learning $\boldsymbol{L}$. Since the generalization error G includes the unknown learning target function $f(\boldsymbol{x})$, we can not directly learn $\boldsymbol{L}$ so that G is minimized. A standard approach to coping with this problem is to employ an accessible estimator of the unknown generalization error G. In the next section, we review existing methods for estimating G.

3 Generalization Error Estimators

In this section, we briefly review generalization error estimators called the *subspace information criterion* (SIC) [20, 19] and its extension the *regularized SIC* (RSIC) [18].

3.1 Subspace Information Criterion

Let $\mathcal{S}$ be a subspace of $\mathcal{H}$ spanned by $\{K(\boldsymbol{x}, \boldsymbol{x}_i)\}_{i=1}^{n}$. Let $g(\boldsymbol{x})$ be the orthogonal projection of $f(\boldsymbol{x})$ onto $\mathcal{S}$. Note that, in the sense of Eq.(4), $g(\boldsymbol{x})$ is the optimal approximation to $f(\boldsymbol{x})$ in $\mathcal{S}$ (see Figure 2). Since $g(\boldsymbol{x})$ belongs to $\mathcal{S}$, it is expressed as

$$g(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i^{*} K(\boldsymbol{x}, \boldsymbol{x}_i), \tag{9}$$

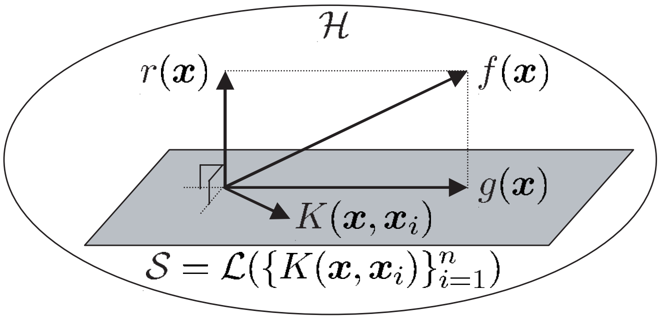

$\mathcal{H}$
$r(\boldsymbol{x})$
$f(\boldsymbol{x})$
$g(\boldsymbol{x})$
$K(\boldsymbol{x},\boldsymbol{x}_i)$
$\mathcal{S}=\mathcal{L}(\{K(\boldsymbol{x},\boldsymbol{x}_i)\}_{i=1}^{n})$

Figure 2: Decomposition of the learning target function $f(\boldsymbol{x})$.

where $\{\alpha_i^*\}_{i=1}^n$ are unknown optimal parameters. Let

$$\boldsymbol{\alpha}^* = (\alpha_1^*, \alpha_2^*, \ldots, \alpha_n^*)^\top. \tag{10}$$

Then the generalization error G can be expressed as follows [19].

$$G[\boldsymbol{L}] = \mathbb{E}_\epsilon \langle \boldsymbol{KLy}, \boldsymbol{Ly} \rangle - 2\mathbb{E}_\epsilon \langle \boldsymbol{KLy}, \boldsymbol{\alpha}^* \rangle, \tag{11}$$

where $\boldsymbol{K}$ is the *kernel matrix*, i.e., the (i,j)-th element is given by

$$\boldsymbol{K}_{i,j} = K(\boldsymbol{x}_i, \boldsymbol{x}_j). \tag{12}$$

Since $\boldsymbol{\alpha}^*$ is unknown in Eq.(11), we replace it by its linear unbiased estimator $\widehat{\boldsymbol{\alpha}}_u$ (see Figure 3). Namely, with some n-dimensional matrix $\boldsymbol{R}_u$, $\widehat{\boldsymbol{\alpha}}_u$ is given as

$$\widehat{\boldsymbol{\alpha}}_u = \boldsymbol{R}_u \boldsymbol{y}, \tag{13}$$

which satisfies

$$\mathbb{E}_\epsilon \widehat{\boldsymbol{\alpha}}_u = \boldsymbol{\alpha}^*. \tag{14}$$

Note that the subscript 'u' in the above equations stands for 'unbiased'. It is known that such $\boldsymbol{R}_u$ is given as follows [19].

$$\boldsymbol{R}_u = \boldsymbol{K}^\dagger, \tag{15}$$

where † denotes the Moore-Penrose generalized inverse [2].

Using $\widehat{\boldsymbol{\alpha}}_u$, we can express G as

$$G[\boldsymbol{L}] = \mathbb{E}_\epsilon \langle \boldsymbol{KLy}, \boldsymbol{Ly} \rangle - 2\mathbb{E}_\epsilon \langle \boldsymbol{KLy}, \boldsymbol{R}_u \boldsymbol{y} \rangle + \sigma^2 \mathrm{tr}(\boldsymbol{KLR}_u^\top). \tag{16}$$

The subspace information criterion (SIC) is defined as the right-hand side of Eq.(16) with the expectation operator $\mathbb{E}_\epsilon$ removed:

$$\mathrm{SIC}[\boldsymbol{L}] = \langle \boldsymbol{KLy}, \boldsymbol{Ly} \rangle - 2\langle \boldsymbol{KLy}, \boldsymbol{R}_u \boldsymbol{y} \rangle + 2\sigma^2 \mathrm{tr}(\boldsymbol{KLR}_u^\top). \tag{17}$$

For any $\boldsymbol{L}$, SIC is an unbiased estimator of G.

$$\mathbb{E}_\epsilon \mathrm{SIC}[\boldsymbol{L}] = G[\boldsymbol{L}]. \tag{18}$$

In the papers [20, 19], the learning matrix $\boldsymbol{L}$ is determined based on SIC by choosing $\boldsymbol{L}$ that minimizes SIC from a set $\mathcal{L}$ of candidates of $\boldsymbol{L}$:

$$\widehat{\boldsymbol{L}} = \mathop{\mathrm{argmin}}_{\boldsymbol{L} \in \mathcal{L}} \mathrm{SIC}[\boldsymbol{L}]. \tag{19}$$

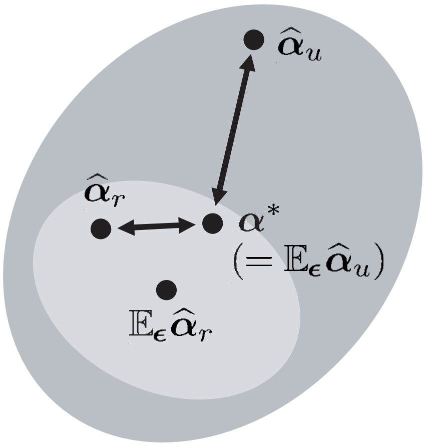

$\widehat{\boldsymbol{\alpha}}_u$
$\widehat{\boldsymbol{\alpha}}_r$
$\boldsymbol{\alpha}^*$
$(= \mathbb{E}_{\boldsymbol{\epsilon}} \widehat{\boldsymbol{\alpha}}_u)$
$\mathbb{E}_{\boldsymbol{\epsilon}} \widehat{\boldsymbol{\alpha}}_r$

Figure 3: Replacing unknown $\boldsymbol{\alpha}^*$ by an unbiased estimator $\widehat{\boldsymbol{\alpha}}_u$ (SIC) or by a regularized estimator $\widehat{\boldsymbol{\alpha}}_r$ (RSIC).

3.2 Regularized Subspace Information Criterion

It is reported that a good learning matrix $\boldsymbol{L}$ can be obtained by SIC [20, 19]. However, the goodness of SIC is only guaranteed in the sense of unbiasedness; nothing has been shown regarding its variance. This implies, e.g., the variance of SIC can be large when the noise level is very high. In such cases, learning with SIC can be unstable. To cope with this problem, the regularized SIC (RSIC) has been proposed [18]. Below, we briefly review RSIC.

Let $\widehat{\boldsymbol{\alpha}}_r$ be some linear regularized estimator of $\boldsymbol{\alpha}^*$:

$$\widehat{\boldsymbol{\alpha}}_r = \boldsymbol{R}\boldsymbol{y}, \tag{20}$$

where $\boldsymbol{R}$ is an n-dimensional matrix which is independent of the noise. We call $\boldsymbol{R}$ the reference matrix, since $\widehat{\boldsymbol{\alpha}}_r$ is used as a reference.

A major reason why SIC can have large variance would be instability of $\widehat{\boldsymbol{\alpha}}_u$. A basic idea of RSIC is to replace the unbiased estimator $\widehat{\boldsymbol{\alpha}}_u$ with a biased but more stable estimator $\widehat{\boldsymbol{\alpha}}_r$ (see Figure 3 again):

$$\mathrm{RSIC}[\boldsymbol{L};\boldsymbol{R}] = \langle \boldsymbol{K}\boldsymbol{L}\boldsymbol{y}, \boldsymbol{L}\boldsymbol{y}\rangle - 2\langle \boldsymbol{K}\boldsymbol{L}\boldsymbol{y}, \boldsymbol{R}\boldsymbol{y}\rangle + 2\sigma^2 \mathrm{tr}(\boldsymbol{K}\boldsymbol{L}\boldsymbol{R}^\top), \tag{21}$$

where the notation $\mathrm{RSIC}[\boldsymbol{L};\boldsymbol{R}]$ means that it is a functional of $\boldsymbol{L}$ with a 'parameter' matrix $\boldsymbol{R}$.

In RSIC, the parameter matrix $\boldsymbol{R}$ should be determined appropriately. To this end, we need a goodness measure of $\boldsymbol{R}$. The paper [18] proposed using the following criterion.

$$J[\boldsymbol{R};\boldsymbol{L}] = \mathbb{E}_{\boldsymbol{\epsilon}}(\mathrm{RSIC}[\boldsymbol{L};\boldsymbol{R}] - G[\boldsymbol{L}])^2, \tag{22}$$

where the notation $J[\boldsymbol{R};\boldsymbol{L}]$ means that it is a functional of $\boldsymbol{R}$ which depends on $\boldsymbol{L}$. Now we want to determine $\boldsymbol{R}$ so that the above J is minimized. However, J includes unknown

G so it can not be directly calculated. Let $\boldsymbol{B}$ and $\boldsymbol{C}$ be

$$\boldsymbol{B} = 2\boldsymbol{R}_u^\top \boldsymbol{K}\boldsymbol{L} - 2\boldsymbol{R}^\top \boldsymbol{K}\boldsymbol{L}, \tag{23}$$

$$\boldsymbol{C} = \boldsymbol{L}^\top \boldsymbol{K}\boldsymbol{L} - 2\boldsymbol{R}^\top \boldsymbol{K}\boldsymbol{L}. \tag{24}$$

Then an unbiased estimator of J is given as follows [18].

$$\begin{aligned}\widehat{J}[\boldsymbol{R};\boldsymbol{L}] = &\left\{\langle \boldsymbol{B}\boldsymbol{y},\boldsymbol{y}\rangle - \sigma^2\mathrm{tr}(\boldsymbol{B})\right\}^2\\ &- \sigma^2\|(\boldsymbol{B}+\boldsymbol{B}^\top)\boldsymbol{y}\|^2 + \sigma^4\mathrm{tr}(\boldsymbol{B}^2+\boldsymbol{B}\boldsymbol{B}^\top)\\ &+ \sigma^2\|(\boldsymbol{C}+\boldsymbol{C}^\top)\boldsymbol{y}\|^2 - \sigma^4\mathrm{tr}(\boldsymbol{C}^2+\boldsymbol{C}\boldsymbol{C}^\top),\end{aligned} \tag{25}$$

which satisfies, for any $\boldsymbol{R}$ and $\boldsymbol{L}$,

$$\mathbb{E}_{\boldsymbol{\epsilon}}\widehat{J}[\boldsymbol{R};\boldsymbol{L}] = J[\boldsymbol{R};\boldsymbol{L}]. \tag{26}$$

The paper [18] proposed using the above $\widehat{J}$ instead of J for determining $\boldsymbol{R}$.

Learning $\boldsymbol{L}$ based on RSIC and $\widehat{J}$ is carried out as follows. First, a set $\mathcal{L}$ of candidates of $\boldsymbol{L}$ and a set $\mathcal{R}$ of candidates of $\boldsymbol{R}$ are prepared. For each $\boldsymbol{L} \in \mathcal{L}$, $\boldsymbol{R}$ is optimized within $\mathcal{R}$:

$$\widehat{\boldsymbol{R}}_{\boldsymbol{L}} = \operatorname*{argmin}_{\boldsymbol{R}\in\mathcal{R}} \widehat{J}[\boldsymbol{R};\boldsymbol{L}]. \tag{27}$$

Then, using $\widehat{\boldsymbol{R}}_{\boldsymbol{L}}$, $\boldsymbol{L}$ is optimized within $\mathcal{L}$:

$$\widehat{\boldsymbol{L}} = \operatorname*{argmin}_{\boldsymbol{L}\in\mathcal{L}} \mathrm{RSIC}[\boldsymbol{L};\widehat{\boldsymbol{R}}_{\boldsymbol{L}}]. \tag{28}$$

4 Existing Methods for Determining $\boldsymbol{L}$

When we learn $\boldsymbol{L}$ using SIC or RSIC, we have to determine the set $\mathcal{L}$ from which $\boldsymbol{L}$ is searched (and also the set $\mathcal{R}$ from which $\boldsymbol{R}$ is searched in RSIC). The largest possible set is $\mathbb{R}^n$, but it is generally too broad to be searched from. Conventionally, we form the set $\mathcal{L}$ (and the set $\mathcal{R}$) based on some learning criterion. In this section, we briefly review popular choices of the learning criterion.

4.1 Existing Method 1 (E1)

Ridge learning [8, 22, 14] determines the parameter $\boldsymbol{\alpha}$ so that the regularized squared error is minimized.

$$\sum_{i=1}^{n}\left(\widehat{f}(\boldsymbol{x}_i) - y_i\right)^2 + \eta\|\boldsymbol{\alpha}\|^2, \tag{29}$$

where η is a non-negative scalar called the ridge parameter. A minimizer of the regularized squared error is given by

$$\boldsymbol{L} = (\boldsymbol{K}^2 + \eta\boldsymbol{I})^{-1}\boldsymbol{K}, \tag{30}$$

where $\boldsymbol{I}$ is the identity matrix. If we focus on ridge learning, the problem of choosing the learning matrix $\boldsymbol{L}$ is reduced to the problem of choosing the ridge parameter η.

The papers [20, 19] proposed determining the learning matrix $\boldsymbol{L}$ by choosing the best value of η that minimizes SIC from a finite set of different values of η. We refer to this procedure as E1. If the computational complexity is measured with respect to the number of compared models, the computational complexity of E1 is $\mathcal{O}(|\mathcal{L}|)$, where $|\mathcal{L}|$ denotes the number of elements in $\mathcal{L}$ (see Table 1).

4.2 Existing Method 2 (E2)

In the procedure E1, the learning matrix $\boldsymbol{L}$ is chosen from a finite set of candidates. In order to improve the optimization quality of $\boldsymbol{L}$, it is desirable to increase the number of candidates. However, increasing the number of candidates simply increases the computational complexity (see Table 1). The paper [21] proposed an efficient model selection procedure based on SIC, where the best learning matrix $\boldsymbol{L}$ is analytically obtained under a certain condition. This analytic approach maximally enhances the optimization quality and at the same time it keeps the computational complexity reasonable.

It appears to be difficult to have an analytic solution of Eq.(27) if the set $\mathcal{L}$ is determined based on ridge learning (30), since the target parameter η is included in the matrix inverse. The paper [21] instead employed *shrinkage learning*: determine the parameter $\boldsymbol{\alpha}$ so that the following quantity is minimized.

$$\sum_{i=1}^{n}\left(\widehat{f}(\boldsymbol{x}_i)-y_i\right)^2+\lambda\|\boldsymbol{K}\boldsymbol{\alpha}\|^2, \tag{31}$$

where λ is a non-negative scalar called the shrinkage parameter. A minimizer of the above quantity is given by the following learning matrix.

$$\boldsymbol{L}=\frac{1}{1+\lambda}\boldsymbol{K}^{\dagger}. \tag{32}$$

If we focus on shrinkage learning, the problem of choosing the learning matrix $\boldsymbol{L}$ is reduced to the problem of choosing the shrinkage parameter λ. Let $\widehat{\lambda}_{\mathrm{SIC}}$ be the minimizer of SIC:

$$\widehat{\lambda}_{\mathrm{SIC}}=\underset{\lambda\in[0,\infty)}{\operatorname{arginf}}\ \mathrm{SIC}(\lambda), \tag{33}$$

and let

$$v_1=\langle\boldsymbol{K}^{\dagger}\boldsymbol{y},\boldsymbol{y}\rangle, \tag{34}$$

$$v_2=\sigma^2\mathrm{tr}(\boldsymbol{K}^{\dagger}). \tag{35}$$

Then $\widehat{\lambda}_{\mathrm{SIC}}$ is given as follows [21]:

$$\widehat{\lambda}_{\mathrm{SIC}}=\begin{cases}\dfrac{v_2}{v_1-v_2} & \text{if } v_1>v_2,\\ \infty & \text{otherwise.}\end{cases} \tag{36}$$

By this expression, we can compute the optimal value of λ analytically. We refer to this procedure as E2. The computational complexity of E2 is $\mathcal{O}(1)$ with respect to the number of candidates (which is infinity here, see Table 1).

4.3 Existing Method 3 (E3)

Although E2 is computationally very efficient, it is based on SIC which can be rather unstable. As explained in Section 3.2, using RSIC would give a more reliable result.

When RSIC is employed, we have to optimize the reference matrix $\boldsymbol{R}$ in addition to the learning matrix $\boldsymbol{L}$. Below, we use ridge learning for both $\boldsymbol{L}$ (see Eq.(30)) and $\boldsymbol{R}$:

$$\boldsymbol{R} = (\boldsymbol{K}^2 + \nu \boldsymbol{I})^{-1}\boldsymbol{K}, \tag{37}$$

where ν is a non-negative scalar. Now the problem of choosing $\boldsymbol{R}$ and $\boldsymbol{L}$ is reduced to the problem of choosing ν and η.

The paper [18] proposed determining the ridge parameter η based on RSIC as follows. First, a finite set of candidate values of η and a finite set of candidate values of ν are prepared. For each η, ν is optimized based on $\widehat{J}$. Then η is optimized based on RSIC using the chosen ν. We refer to this procedure as E3. The computational complexity of E3 is $\mathcal{O}(|\mathcal{L}||\mathcal{R}|)$ (see Table 1).

The procedure E3 has been shown to work well [18]. However, as Table 1 shows, it is computationally demanding. The primal goal of this paper is to give a more efficient model optimization procedure based on RSIC.

5 Efficient Optimization of $\boldsymbol{R}$

In this section, we present a method to analytically obtain the best reference matrix $\boldsymbol{R}$ under a certain condition. This analytic approach maximally enhances the optimization quality while the computational cost is kept moderate.

5.1 Analytic Expression of Optimal $\boldsymbol{R}$

Let us employ shrinkage learning for $\boldsymbol{R}$:

$$\boldsymbol{R} = \frac{1}{1+\gamma}\boldsymbol{K}^{\dagger}, \tag{38}$$

where γ is a non-negative scalar. Here we derive an analytic form of the optimal γ that minimizes $\widehat{J}$. Note that we do not impose any assumption on $\boldsymbol{L}$. That is, the following discussion is valid for any $\boldsymbol{L}$.

Let $\boldsymbol{S}$ and $\boldsymbol{T}$ be

$$\boldsymbol{S} = \boldsymbol{K}^{\dagger}\boldsymbol{K}\boldsymbol{L}, \tag{39}$$

$$\boldsymbol{T} = \boldsymbol{L}^{\top}\boldsymbol{K}\boldsymbol{L}. \tag{40}$$

Let

$$\begin{aligned} u_1 &= \left\{\langle \boldsymbol{S}\boldsymbol{y}, \boldsymbol{y}\rangle - \sigma^2 \mathrm{tr}(\boldsymbol{S})\right\}^2, \qquad (41) \\ u_2 &= \sigma^2 \|(\boldsymbol{S} + \boldsymbol{S}^\top)\boldsymbol{y}\|^2 - \sigma^4 \mathrm{tr}(\boldsymbol{S}^2 + \boldsymbol{S}\boldsymbol{S}^\top) \\ &\quad - \sigma^2 \langle (\boldsymbol{S} + \boldsymbol{S}^\top)\boldsymbol{T}\boldsymbol{y}, \boldsymbol{y}\rangle + \sigma^4 \mathrm{tr}(\boldsymbol{S}\boldsymbol{T}). \qquad (42) \end{aligned}$$

Then we have the following theorem.

Theorem 1 *Let*

$$\widehat{\gamma}_{\boldsymbol{L}} = \operatorname*{arginf}_{\gamma \in [0,\infty)} \widehat{J}(\gamma; \boldsymbol{L}). \qquad (43)$$

Then $\widehat{\gamma}_{\boldsymbol{L}}$ *is given by*

$$\widehat{\gamma}_{\boldsymbol{L}} = \begin{cases} \max\left(0, \dfrac{u_2}{u_1 - u_2}\right) & \textit{if } u_1 > u_2, \\ \textit{arbitrary value in } [0, \infty) & \textit{if } u_1 = u_2 = 0, \\ \infty & \textit{otherwise.} \end{cases} \qquad (44)$$

A proof of the above theorem is given in A. By Theorem 1, the optimal value of γ can be analytically calculated for any $\boldsymbol{L}$. Note that the second case in Eq.(44) may not occur in practice.

5.2 Proposed Method 1 (P1)

To be comparable to E3, we use ridge learning for $\boldsymbol{L}$ on top of Theorem 1. That is, we prepare a finite set of candidate values of η, and for each η, the best γ is computed by Eq.(44). We refer to this procedure as P1. The computational complexity of P1 is $\mathcal{O}(|\mathcal{L}|)$ which is smaller than E3 by the factor of $|\mathcal{R}|$. This order is comparable to E1 (see Table 1).

6 Efficient Optimization of $\boldsymbol{L}$

In the previous section, we derived an analytic expression of $\boldsymbol{R}$, which contributes to reducing the computation time. In this section, we show that we can even derive an analytic expression of the optimal $\boldsymbol{L}$ under a certain condition, which further improves the computational cost.

6.1 Analytic Expression of Optimal $\boldsymbol{L}$

We employ shrinkage learning also for $\boldsymbol{L}$ (see Eq.(32)). Here we derive the analytic form of the optimal shrinkage parameter λ on top of Theorem 1.

Let

$$v_3 = 2\sigma^2 \langle (\boldsymbol{K}^\dagger)^2 \boldsymbol{y}, \boldsymbol{y}\rangle - \sigma^4 \mathrm{tr}((\boldsymbol{K}^\dagger)^2). \qquad (45)$$

Then we have the following theorem.

Theorem 2 *Let $\widehat{\gamma}_\lambda$ be the optimal γ for Eq.*(32) *(see Theorem 1), and let*

$$\widehat{\lambda}_{\mathrm{RSIC}} = \operatorname*{arginf}_{\gamma\in[0,\infty)} \mathrm{RSIC}(\lambda;\widehat{\gamma}_\lambda). \tag{46}$$

Then $\widehat{\lambda}_{\mathrm{RSIC}}$ is given by

$$\widehat{\lambda}_{\mathrm{RSIC}} = \begin{cases} \dfrac{(v_1-v_2)v_2}{(v_1-v_2)^2 - 2\max(0,v_3)} & \textit{if } v_1 > v_2 \textit{ and } v_3 < \frac{(v_1-v_2)^2}{2}, \\ \textit{arbitrary value in } [0,\infty) & \textit{if } v_1 = v_2 = 0, \\ \infty & \textit{otherwise}, \end{cases} \tag{47}$$

where v_1 and v_2 are defined by Eqs.(34) *and* (35)*, respectively.*

A proof of the above theorem is given in B. The proof is rather elaborate since we have to take into account λ which implicitly appears in $\widehat{\gamma}_\lambda$ (see Eq.(44)). Fortunately, however, we could have obtained a rather simple formula for $\widehat{\lambda}_{\mathrm{RSIC}}$. By Theorem 2, the optimal value of λ can be analytically calculated. Note that the second case in Eq.(47) may not occur in practice.

The above theorem may be regarded as an extension of E2 where the optimal shrinkage parameter is derived for SIC. It is interesting to note that the first case in Eq.(47) with $v_3 \leq 0$ agrees with the first case in Eq.(36).

6.2 Proposed Method 2 (P2)

As P2, we refer to the procedure of obtaining $\boldsymbol{L}$ by Theorem 2. The computational complexity of P2 is $\mathcal{O}(1)$, which is smaller than P1 by the factor of $|\mathcal{L}|$. This order is comparable to E2 (see Table 1).

7 Simulations

In this section, we experimentally compare the accuracy and computation time of the existing and proposed methods.

7.1 Setting

Let $f(x)$ be

$$f(x) = \mathrm{sinc}(x). \tag{48}$$

See Figure 4 for the profile. We employ the Gaussian reproducing kernel Hilbert space [15] as $\mathcal{H}$, where the reproducing kernel is given by

$$K(x,x') = \exp\left(-\frac{(x-x')^2}{2}\right). \tag{49}$$

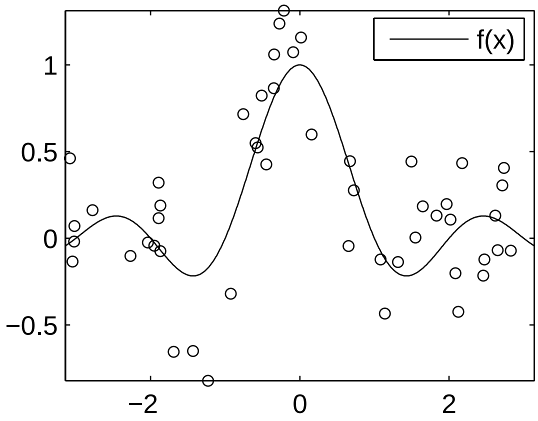


Figure 4: Sinc function and training samples when $(n, \sigma^2) = (50, 0.01)$.

Note that the sinc function is included in the above Gaussian reproducing kernel Hilbert space [5]. We take training input points $\{x_i\}_{i=1}^n$ independently following the uniform distribution on $(-\pi, \pi)$. Noise $\{\epsilon_i\}_{i=1}^n$ are taken independently following the normal distribution with mean zero and variance σ^2. Training output values $\{y_i\}_{i=1}^n$ are created as $y_i = \mathrm{sinc}(x_i) + \epsilon_i$. We consider the following four cases.

$$\begin{aligned}(n, \sigma^2) =&(50, 0.01), (50, 0.09),\\ &(100, 0.01), (100, 0.09).\end{aligned} \tag{50}$$

That is, small/large samples and low/high noise level. For each of the above case, we repeat the simulation 1000 times by changing $\{x_i\}_{i=1}^n$ and $\{\epsilon_i\}_{i=1}^n$. In the experiments, σ^2 is treated as an unknown variable and is estimated by

$$\widehat{\sigma^2} = \frac{\|\boldsymbol{K}\boldsymbol{K}^\dagger \boldsymbol{y} - \boldsymbol{y}\|^2}{n - \mathrm{tr}(\boldsymbol{K}\boldsymbol{K}^\dagger)}. \tag{51}$$

Note that the current setting theoretically yields $\boldsymbol{K}\boldsymbol{K}^\dagger = \boldsymbol{I}$ with probability one since the Gaussian kernel with distinct training input points provides a strictly positive kernel matrix [15]. However, in practice, $\boldsymbol{K}$ is degenerated numerically so Eq.(51) is still valid.

η and ν are chosen from the set of 10 equidistance values in log-scale in the range $[10^{-4}, 10^4]$. Therefore, $|\mathcal{L}| = |\mathcal{R}| = 10$. In our implementation, the computation of Moore-Penrose generalized inverse was rather unstable. To avoid numerical troubles, we discarded eigenvalues less than 10^{-2}.

So far, we called G the generalization error, where $\|f\|^2$ is ignored (see Eq.(3)). This was convenient when we investigate relative goodness of $\widehat{f}$. However in the experiments, we are interested in absolute goodness of $\widehat{f}$. So we use the following measure here.

$$\overline{G} = \mathbb{E}_\epsilon \|\widehat{f} - f\|^2 = G + \|f\|^2. \tag{52}$$

With some abuse, we call $\overline{G}$ the generalization error through this section.

Note that all matrices $\boldsymbol{L}$, $\boldsymbol{R}$, and $\boldsymbol{K}$ appeared in the current setting have common eigenvectors. This means that all the methods can be implemented quite efficiently, i.e., once eigendecomposition of $\boldsymbol{K}$ is carried out in advance, all the methods can be computed very efficiently. We implemented all the methods in this way.

In the following, we compare the computation time and the generalization error obtained by E1, E2, E3, P1, and P2.

7.2 Overview of the Results

Mean and standard deviation of the generalization error obtained by each method over 1000 runs are described in Table 2, where the best method and comparable ones by the *t-test* [7] at the significance level 5% are described with boldface. Figure 5 shows the box-plot expression of the obtained generalization error. The table shows that E3 works significantly better than others when $n = 50$, while P1 gives the best performance when $n = 100$. Mean CPU computation time over 1000 trials is described in Table 3, showing that E3—which appeared to work very well—is slow in computation compared with others.

The ratio of mean generalization error for some pairs of methods is described in Table 4, while the ratio of computation time is described in Table 5. In the following, we compare the pairs in detail.

7.3 Comparison between P1 and E1

The computational complexity of P1 and E1 are both $\mathcal{O}(|\mathcal{L}|)$. We first compare their actual computation time. Table 5 shows that although they have the same computational complexity, P1 required approximately three times more computation time than E1. We conjecture that this is mainly caused by the computation of Eq.(44).

As explained in Section 3.2, model selection by RSIC is expected to be better than that by SIC. Therefore, we expect that P1 gives smaller generalization error than E1, which is investigated experimentally. Table 4 shows that except for $(n, \sigma^2) = (50, 0.01)$, P1 is significantly better than E1. Figure 5 shows that P1 particularly gives smaller upper quantiles than E1. This can also be confirmed from Table 2, where the standard deviation of P1 is much smaller than that of E1.

The above results show that P1 needs slightly longer computation time than E1, but is more accurate and particularly stable than E1.

7.4 Comparison between P1 and E3

P1 and E3 both employ RSIC for model selection. The computational complexity of E3 is $\mathcal{O}(|\mathcal{L}||\mathcal{R}|)$ while that of P1 is $\mathcal{O}(|\mathcal{L}|)$. Thus P1 is theoretically 10 times faster than E3 in computation. Table 5 shows that for both $n = 50$ and $n = 100$, the computation time of P1 is 12–13% of that of E3. Given there are some inessential computations in

Table 1: Computational complexity of each method with respect to the number of compared models. $|\mathcal{L}|$ and $|\mathcal{R}|$ denote the number of elements in $\mathcal{L}$ and $\mathcal{R}$, respectively.

E1	E2	E3	P1	P2
$\mathcal{O}(\|\mathcal{L}\|)$	$\mathcal{O}(1)$	$\mathcal{O}(\|\mathcal{L}\|\|\mathcal{R}\|)$	$\mathcal{O}(\|\mathcal{L}\|)$	$\mathcal{O}(1)$

Table 2: Mean and standard deviation of generalization error for toy data set. The best method and comparable ones by the t-test at the significance level 5% are described with boldface.

(n, σ^2)	E1	E2	E3	P1	P2
$(50, 0.01)$	0.97±0.42	1.16±0.60	**0.92±0.36**	1.11±0.34	1.16±0.58
$(50, 0.09)$	3.06±3.89	4.86±3.86	**2.28±1.84**	2.81±0.83	4.84±2.72
$(100, 0.01)$	1.00±0.50	1.45±0.91	0.88±0.39	**0.85±0.09**	1.44±0.88
$(100, 0.09)$	3.67±4.78	6.40±6.35	2.22±2.84	**1.49±0.65**	5.66±4.54

Table 3: Mean CPU computation time in milli seconds.

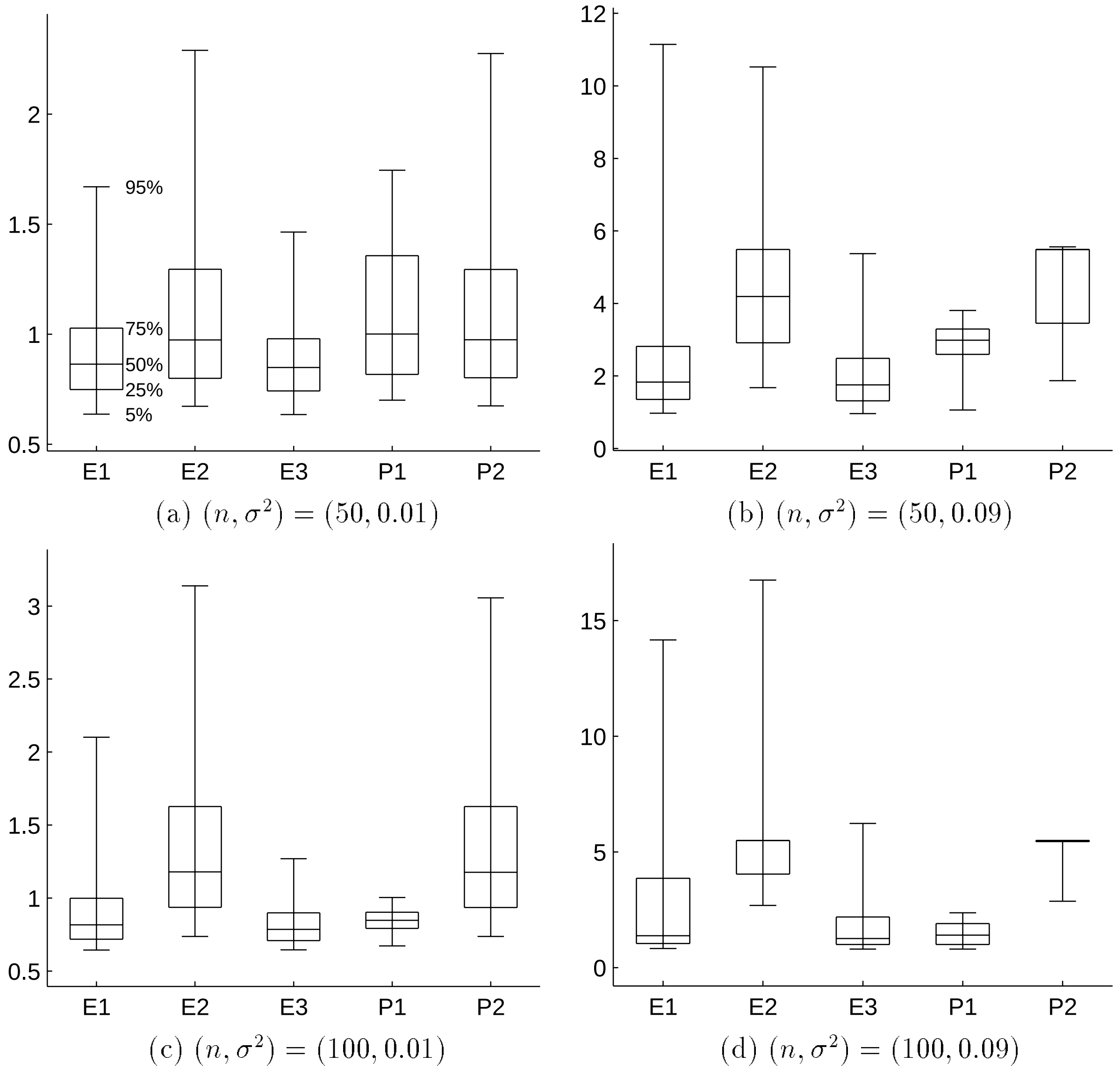
	E1	E2	E3	P1	P2
Theoretical	10	1	100	10	1
$n = 50$	0.32	0.11	7.60	0.95	0.12
$n = 100$	0.38	0.13	8.34	1.05	0.14

Table 4: Ratio of Mean Generalization error. We described the number with bold face if the compared methods have significant difference by the t-test at the significance level 5%.

(n, σ^2)	P1/E1	P1/E3	P2/E2	P2/P1
$(50, 0.01)$	**1.14**	**1.20**	1.00	**1.05**
$(50, 0.09)$	**0.91**	**1.23**	1.00	**1.73**
$(100, 0.01)$	**0.85**	**0.97**	0.99	**1.69**
$(100, 0.09)$	**0.41**	**0.67**	**0.89**	**3.80**

Table 5: Ratio of Mean CPU computation time

	P1/E1	P1/E3	P2/E2	P2/P1
Theoretical	1	0.1	1	0.1
$n = 50$	2.97	0.12	1.09	0.13
$n = 100$	2.76	0.13	1.08	0.13



(a) $(n, \sigma^2) = (50, 0.01)$

(b) $(n, \sigma^2) = (50, 0.09)$

(c) $(n, \sigma^2) = (100, 0.01)$

(d) $(n, \sigma^2) = (100, 0.09)$

Figure 5: Boxplots of generalization error for toy data set.

actual implementation, this experimental result is in good agreement with the theoretical analysis.

E3 employs ridge learning for $\boldsymbol{R}$ while P1 employs shrinkage learning for $\boldsymbol{R}$. Therefore, E3 and P1 generally give different learning results. We experimentally investigate the accuracy of learning. Table 4 shows that, when $n = 50$, the generalization error obtained by P1 is approximately 20% larger than that obtained by E3. On the other hand, when $n = 100$, the generalization error obtained by P1 is smaller than that obtained by E3. Particularly when $(n, \sigma^2) = (100, 0.09)$, P1 prominently outperforms E3 (see also Table 2).

The above results show that P1 is much faster than E3, and the generalization performance may be comparable.

7.5 Comparison between P2 and E2

The computational complexity of P2 and E2 are both $\mathcal{O}(1)$. Table 5 shows that their actual computation time is certainly comparable.

P2 employs RSIC while E2 uses SIC, from which we expect that P2 works better than E2. Table 4 shows that when $\sigma^2 = 0.01$, P2 and E2 do not have statistically significant difference in generalization performance. This would be a natural consequence since when the noise level is low, SIC is already a good estimator of the generalization error without any modification. When $(n, \sigma^2) = (100, 0.09)$, P2 gives significantly better results than E2, which may be caused by the fact that RSIC is a more stable estimator of the generalization error than SIC. When $(n, \sigma^2) = (50, 0.09)$, the mean generalization errors of P2 and E2 are quite comparable. However, Figure 5 shows that P2 gives much smaller 95%-quantile than E2, which implies that P2 is more stable than E2. This can also be confirmed from Table 2, where the standard deviation of P2 is smaller than that of E2.

The above results show that P2 is on par with E2 in computation time, but is more stable than E2 when the noise level is high.

7.6 Comparison between P2 and P1

Finally, we compare two proposed methods. The computational complexity of P1 is $\mathcal{O}(|\mathcal{L}|)$, while that of P2 is $\mathcal{O}(1)$. Thus P2 is theoretically 10 times faster than E1 in computation. Table 5 shows that for both $n = 50$ and $n = 100$, the computation time of P2 is 13% of that of E3, which is in good agreement with the theoretical analysis.

P1 employs ridge learning for $\boldsymbol{L}$ while P2 uses shrinkage learning for $\boldsymbol{L}$. Therefore, P1 and P2 generally give different learning results. Table 4 shows that for all four cases, P1 gives significantly smaller generalization errors than P2; particularly the difference is prominent when $\sigma^2 = 0.09$.

The above results show that P2 is faster in computation than P1, but P1 works better in generalization performance than P2.

7.7 Summary

The above experimental results are summarized as follows. When we do not have limitation in computing time and just pursue the optimal generalization capability, E3 or P1 appears to be suitable. Among them, P1 is faster in computation than E3. For this reason, we conclude that P1 is the most promising method in pursuing the optimal generalization capability.

On the other hand, if we have a limit in computation time, P2 seems to be the best choice since it gives the best generalization performance among the class of computationally efficient methods.

8 Conclusions

An existing model selection procedure based on RSIC chooses $\boldsymbol{L}$ and $\boldsymbol{R}$ from finite sets, i.e., a grid search. If the number of candidates in the sets are increased, the optimization quality of $\boldsymbol{L}$ and $\boldsymbol{R}$ would be improved. However, this in turn increases the computation time. In this paper, we alleviate this problem by deriving analytic expressions of the optimal $\boldsymbol{L}$ and $\boldsymbol{R}$ from infinite sets. We experimentally showed that the proposed model selection procedures based on the analytic optimal solutions are more efficient than existing methods.

In deriving analytic solutions, we used shrinkage learning since it has a simple expression. However, it appears to be rather inaccurate compared with ridge learning. Our important future work is to derive analytic expressions of the optimal expressions of $\boldsymbol{L}$ and $\boldsymbol{R}$ under a more powerful learning criterion.

Acknowledgments

The authors would like to thank Hidemitsu Ogawa for valuable comments. We acknowledge financial support from MEXT (Grant-in-Aid for Young Scientists 17700142).

A Proof of Theorem 1

When $\boldsymbol{R}$ is of the form of Eq.(38), $\boldsymbol{B}$ and $\boldsymbol{C}$ are expressed as

$$\boldsymbol{B} = \frac{2\gamma}{1+\gamma}\boldsymbol{S}, \tag{53}$$

$$\boldsymbol{C} = \boldsymbol{T} - \frac{2}{1+\gamma}\boldsymbol{S}. \tag{54}$$

Let

$$u_0 = \sigma^2\|(\boldsymbol{S}+\boldsymbol{S}^\top)\boldsymbol{y}\|^2 - \sigma^4\mathrm{tr}(\boldsymbol{S}^2+\boldsymbol{S}\boldsymbol{S}^\top). \tag{55}$$

Then Eq.(25) yields

$$\begin{aligned}\widehat{J}[\gamma;\boldsymbol{L}] =& \frac{4(u_1-u_0)\gamma^2+4u_0}{(1+\gamma)^2} + \frac{8(u_2-u_0)}{1+\gamma}\\ &+ 4\sigma^2\|\boldsymbol{T}\boldsymbol{y}\|^2 - 2\sigma^4\mathrm{tr}(\boldsymbol{T}^2),\end{aligned} \tag{56}$$

and its first derivative is given by

$$\widehat{J}'[\gamma;\boldsymbol{L}] = \frac{8\{(u_1-u_2)\gamma - u_2\}}{(1+\gamma)^3}. \tag{57}$$

Below, we give a proof depending on u_1 and u_2 (See Table 6).

Table 6: Cases in Proof of Theorem 1.

Conditions		Results
$u_1 > u_2$	$u_2 < 0$	(A) $\widehat{\gamma}_{\boldsymbol{L}} = 0$
	$u_2 \geq 0$	(B) $\widehat{\gamma}_{\boldsymbol{L}} = \widetilde{\gamma}$
$u_1 < u_2$		(C) $\widehat{\gamma}_{\boldsymbol{L}} = \infty$
$u_1 = u_2$	$u_2 \neq 0$	(D) $\widehat{\gamma}_{\boldsymbol{L}} = \infty$
	$u_2 = 0$	(E) $\widehat{\gamma}_{\boldsymbol{L}} \in [0, \infty)$

(A) If $u_1 > u_2$ and $u_2 < 0$, Eq.(57) yields $\widehat{J}'[\gamma; \boldsymbol{L}] > 0$ for any $\gamma \in [0, \infty)$. This implies that $\widehat{J}[\gamma; \boldsymbol{L}]$ is monotone increasing and thus $\widehat{\gamma}_{\boldsymbol{L}} = 0$.

(B) If $u_1 > u_2 \geq 0$, Eq.(57) implies that $\widehat{J}'[\widetilde{\gamma}; \boldsymbol{L}] = 0$, where

$$\widetilde{\gamma} = \frac{u_2}{u_1 - u_2} \ (\geq 0). \tag{58}$$

Since

$$\widehat{J}[\gamma; \boldsymbol{L}] - \widehat{J}[\widetilde{\gamma}; \boldsymbol{L}] = \frac{4u_1(\gamma - \widetilde{\gamma})^2}{(1+\gamma)^2(1+\widetilde{\gamma})^2} \geq 0, \tag{59}$$

where strict equality holds if and only if $\gamma = \widetilde{\gamma}$, we have $\widehat{\gamma}_{\boldsymbol{L}} = \widetilde{\gamma}$.

(C) If $u_1 < u_2$, we have $u_2 > 0$ since $u_1 \geq 0$. Then Eq.(57) yields $\widehat{J}'[\gamma; \boldsymbol{L}] < 0$ for any $\gamma \in [0, \infty)$. This implies that $\widehat{J}[\gamma; \boldsymbol{L}]$ is monotone decreasing and thus $\widehat{\gamma}_{\boldsymbol{L}} = \infty$.

(D) If $u_1 = u_2 \neq 0$, we have $u_2 > 0$ since $u_1 > 0$. Then Eq.(57) yields $\widehat{J}'[\gamma; \boldsymbol{L}] < 0$ for any $\gamma \in [0, \infty)$. This implies that $\widehat{J}[\gamma; \boldsymbol{L}]$ is monotone decreasing and thus $\widehat{\gamma}_{\boldsymbol{L}} = \infty$.

(E) If $u_1 = u_2 = 0$, Eq.(57) yields $\widehat{J}'[\gamma; \boldsymbol{L}] = 0$. This implies that $\widehat{J}[\gamma; \boldsymbol{L}]$ is constant so $\widehat{\gamma}_{\boldsymbol{L}}$ is an arbitrary value in $[0, \infty)$.

By summarizing the above results (see Table 6), we have Eq.(44). ■

B Proof of Theorem 2

When $\boldsymbol{L}$ is of the form of Eq.(32), $\boldsymbol{S}$ and $\boldsymbol{T}$ are expressed as

$$\boldsymbol{S} = \frac{1}{1+\lambda}\boldsymbol{K}^{\dagger}, \tag{60}$$

$$\boldsymbol{T} = \frac{1}{(1+\lambda)^2}\boldsymbol{K}^{\dagger}. \tag{61}$$

Then we have

$$u_2 = \frac{v_3(1+2\lambda)}{(1+\lambda)^3}, \tag{62}$$

$$u_1 - u_2 = \frac{(v_1 - v_2)^2(1+\lambda) - v_3(1+2\lambda)}{(1+\lambda)^3}, \tag{63}$$

$$\mathrm{RSIC}[\lambda; \widehat{\gamma}_{\lambda}] = \frac{v_1}{(1+\lambda)^2} - \frac{2(v_1 - v_2)}{(1+\lambda)(1+\widehat{\gamma}_{\lambda})}. \tag{64}$$

Table 7: Cases in Proof of Theorem 2.

Conditions			Results
$v_1 \neq v_2$	$v_3 \leq 0$	$v_1 < v_2$	(A1) $\widehat{\lambda}_{\mathrm{RSIC}} = \infty$
		$v_1 > v_2$	(A2) $\widehat{\lambda}_{\mathrm{RSIC}} = \widetilde{\lambda}$
	$0 < v_3 < \frac{(v_1-v_2)^2}{2}$	$v_1 < v_2$	(B1) $\widehat{\lambda}_{\mathrm{RSIC}} = \infty$
		$v_1 > v_2$	(B2) $\widehat{\lambda}_{\mathrm{RSIC}} = \overline{\lambda}$
	$v_3 = \frac{(v_1-v_2)^2}{2}$	$v_2 > 0$	(B3) $\widehat{\lambda}_{\mathrm{RSIC}} = \infty$
		$v_2 = 0$	(B4) Do not happen
	$\frac{(v_1-v_2)^2}{2} < v_3 < (v_1 - v_2)^2$	$v_1 > 0$	(C1) $\widehat{\lambda}_{\mathrm{RSIC}} = \infty$
		$v_1 = 0$	(C2) Do not happen
	$(v_1 - v_2)^2 \leq v_3$	$v_1 > 0$	(D1) $\widehat{\lambda}_{\mathrm{RSIC}} = \infty$
		$v_1 = 0$	(D2) Do not happen
$v_1 = v_2$	$v_1 = 0$		(E) $\widehat{\lambda}_{\mathrm{RSIC}} \in [0, \infty)$
	$v_1 > 0$		(F) $\widehat{\lambda}_{\mathrm{RSIC}} = \infty$

Since $\boldsymbol{K}$ is positive semidefinite, we have $v_1 \geq 0$ and $v_2 \geq 0$. Below, we give a proof depending on v_1, v_2, and v_3 (See Table 7).

(A) If $v_1 \neq v_2$ and $v_3 \leq 0$, Eq.(62) yields $u_2 < 0$ for any $\lambda \in [0, \infty)$ and Eq.(63) yields $u_1 > u_2$ for any $\lambda \in [0, \infty)$. Therefore, Theorem 1 yields $\widehat{\gamma}_\lambda = 0$ for any $\lambda \in [0, \infty)$. In this case, Eq.(64) yields

$$\mathrm{RSIC}[\lambda; 0] = \frac{-2(v_1 - v_2)\lambda - v_1 + 2v_2}{(1+\lambda)^2}, \tag{65}$$

and its first derivative is given by

$$\mathrm{RSIC}'[\lambda; 0] = \frac{2(v_1 - v_2)\lambda - 2v_2}{(1+\lambda)^3}. \tag{66}$$

(A1) If $v_3 \leq 0$ and $v_1 < v_2$, Eq.(66) yields $\mathrm{RSIC}'[\lambda; 0] < 0$ for any $\lambda \in [0, \infty)$ since $v_2 > 0$ because of $v_2 > v_1 \geq 0$. This implies that $\mathrm{RSIC}[\lambda; 0]$ is monotone decreasing and thus $\widehat{\lambda}_{\mathrm{RSIC}} = \infty$. (A2) If $v_3 \leq 0$ and $v_1 > v_2$, Eq.(66) yields $\mathrm{RSIC}'[\widetilde{\lambda}; 0] = 0$, where

$$\widetilde{\lambda} = \frac{v_2}{v_1 - v_2}. \tag{67}$$

Since

$$\mathrm{RSIC}[\lambda; 0] - \mathrm{RSIC}[\widetilde{\lambda}; 0] = \frac{(v_1 - v_2)^2(\lambda - \widetilde{\lambda})^2}{v_1(1+\lambda)^2} \geq 0, \tag{68}$$

where strict equality holds if and only if $\lambda = \widetilde{\lambda}$, we have $\widehat{\lambda}_{\mathrm{RSIC}} = \widetilde{\lambda}$.

(B) If $v_1 \neq v_2$ and $0 < v_3 \leq \frac{(v_1-v_2)^2}{2}$, Eq.(62) yields $u_2 > 0$ for any $\lambda \in [0, \infty)$ and Eq.(63) yields $u_1 > u_2$ for any $\lambda \in [0, \infty)$. Therefore, Theorem 1 yields $\widehat{\gamma}_\lambda = \widetilde{\gamma}$, where $\widetilde{\gamma}$

is defined by Eq.(58). In this case, Eq.(64) yields

$$\mathrm{RSIC}[\lambda;\widetilde{\gamma}] = \frac{-2v_0\lambda + v_2 - v_0}{(1+\lambda)^2}, \tag{69}$$

where

$$v_0 = \frac{(v_1 - v_2)^2 - 2v_3}{v_1 - v_2}. \tag{70}$$

Then its first derivative is given by

$$\mathrm{RSIC}'[\lambda;\widetilde{\gamma}] = \frac{2v_0\lambda - 2v_2}{(1+\lambda)^3}. \tag{71}$$

(B1) If $0 < v_3 < \frac{(v_1-v_2)^2}{2}$ and $v_1 < v_2$, we have $v_0 < 0$ and $v_2 > 0$ so Eq.(71) yields $\mathrm{RSIC}'[\lambda;\widetilde{\gamma}] < 0$ for any $\lambda \in [0,\infty)$. This implies that $\mathrm{RSIC}[\lambda;\widetilde{\gamma}]$ is monotone decreasing and thus $\widehat{\lambda}_{\mathrm{RSIC}} = \infty$. (B2) If $0 < v_3 < \frac{(v_1-v_2)^2}{2}$ and $v_1 > v_2$, we have $v_0 > 0$ so Eq.(71) yields $\mathrm{RSIC}'[\overline{\lambda};\widetilde{\gamma}] = 0$, where

$$\overline{\lambda} = \frac{v_2}{v_0}\ (\geq 0). \tag{72}$$

Since

$$\mathrm{RSIC}[\lambda;\widetilde{\gamma}] - \mathrm{RSIC}[\overline{\lambda};\widetilde{\gamma}] = \frac{v_0(\lambda - \overline{\lambda})^2}{(1+\lambda)^2(1+\overline{\lambda})} \geq 0, \tag{73}$$

where strict equality holds if and only if $\lambda = \overline{\lambda}$, we have $\widehat{\lambda}_{\mathrm{RSIC}} = \overline{\lambda}$. (B3) If $v_1 \neq v_1$, $v_3 = \frac{(v_1-v_2)^2}{2}$, and $v_2 > 0$, Eq.(71) yields $\mathrm{RSIC}'[\lambda;\widetilde{\gamma}] < 0$ for any $\lambda \in [0,\infty)$. This implies that $\mathrm{RSIC}[\lambda;\widetilde{\gamma}]$ is monotone decreasing and thus $\widehat{\lambda}_{\mathrm{RSIC}} = \infty$. (B4) $v_1 \neq v_1$, $v_3 = \frac{(v_1-v_2)^2}{2}$, and $v_2 = 0$ do not happen; $v_3 \neq 0$ implies $\sigma^2 > 0$, so $v_2 = 0$ yields $\mathrm{tr}(\boldsymbol{K}^\dagger) = 0$ which results in $v_1 = 0$. However, this contradicts with $v_1 \neq v_2$.

(C) If $v_1 \neq v_2$ and $\frac{(v_1-v_2)^2}{2} < v_3 < (v_1 - v_2)^2$, Eq.(62) yields $u_2 > 0$ and Eq.(63) yields $u_1 > u_2$ for any $\lambda \in [0, \lambda_C)$, where

$$\lambda_C = \frac{v_3 - (v_1 - v_2)^2}{(v_1 - v_2)^2 - 2v_3}. \tag{74}$$

Therefore, Theorem 1 yields $\widehat{\gamma}_\lambda = \widetilde{\gamma}$ for $\lambda \in [0, \lambda_C)$. In this case, $\mathrm{RSIC}[\lambda;\widetilde{\gamma}]$ is given by Eq.(69) and $\mathrm{RSIC}'[\lambda;\widetilde{\gamma}]$ is given by Eq.(71). For $\lambda \in [0,\lambda_C)$, we have $\mathrm{RSIC}'[\lambda;\widetilde{\gamma}] < 0$. Therefore, $\mathrm{RSIC}[\lambda;\widetilde{\gamma}]$ is monotone decreasing in $\lambda \in [0, \lambda_C)$, so the infimum of $\mathrm{RSIC}[\lambda;\widetilde{\gamma}]$ in $[0, \lambda_C)$ is

$$\inf_{\lambda \in [0,\lambda_C)} \mathrm{RSIC}[\lambda;\widetilde{\gamma}] = \mathrm{RSIC}[\lambda_C;\widetilde{\gamma}] = \frac{v_1}{(1+\lambda)^2}. \tag{75}$$

Eq.(63) yields $u_1 < u_2$ for any $\lambda \in [\lambda_C, \infty)$. Therefore, Theorem 1 yields $\widehat{\gamma}_\lambda = \infty$ for $\lambda \in [\lambda_C, \infty)$. In this case, Eq.(64) yields

$$\mathrm{RSIC}[\lambda;\infty] = \frac{v_1}{(1+\lambda)^2}, \tag{76}$$

and its first derivative is given by

$$\mathrm{RSIC}'[\lambda;\infty] = -\frac{2v_1}{(1+\lambda)^3}. \tag{77}$$

(C1) If $v_1 \neq v_2$, $\frac{(v_1-v_2)^2}{2} < v_3 < (v_1-v_2)^2$, and $v_1 > 0$, Eq.(77) yields $\mathrm{RSIC}'[\lambda;\infty] < 0$ for any $\lambda \in [\lambda_C, \infty)$. This implies that $\mathrm{RSIC}[\lambda;\infty]$ is monotone decreasing, so the infimum of $\mathrm{RSIC}[\lambda;\infty]$ in $[\lambda_C,\infty)$ is

$$\inf_{\lambda\in[\lambda_C,\infty)} \mathrm{RSIC}[\lambda;\infty] = \mathrm{RSIC}[\infty;\infty] = 0. \tag{78}$$

On the other hand, Eq.(75) yields

$$\inf_{\lambda\in[0,\lambda_C)} \mathrm{RSIC}[\lambda;\widetilde{\gamma}] > 0, \tag{79}$$

from which we have $\widehat{\lambda}_{\mathrm{RSIC}} = \infty$. (C2) $v_1 \neq v_2$, $\frac{(v_1-v_2)^2}{2} < v_3 < (v_1-v_2)^2$, and $v_1 = 0$ do not happen; $v_1 = 0$ implies $\boldsymbol{K}^\dagger\boldsymbol{y} = \boldsymbol{0}$, so $\langle(\boldsymbol{K}^\dagger)^2\boldsymbol{y},\boldsymbol{y}\rangle = 0$, which yields $v_3 = -\sigma^4\mathrm{tr}((\boldsymbol{K}^\dagger)^2) \leq 0$. However, this contradicts with $\frac{(v_1-v_2)^2}{2} < v_3 < (v_1-v_2)^2$.

(D) If $v_1 \neq v_2$ and $(v_1-v_2)^2 \leq v_3$, Eq.(63) yields $u_1 < u_2$ for any $\lambda \in [0,\infty)$. Therefore, Theorem 1 yields $\widehat{\gamma}_\lambda = \infty$. In this case, $\mathrm{RSIC}[\lambda;\infty]$ is given by Eq.(76) and $\mathrm{RSIC}'[\lambda;\infty]$ is given by Eq.(77). (D1) If $v_1 \neq v_2$, $(v_1-v_2)^2 \leq v_3$, and $v_1 > 0$, Eq.(77) yields $\mathrm{RSIC}'[\lambda;\infty] < 0$ for any $\lambda \in [0,\infty)$. Therefore, $\mathrm{RSIC}[\lambda;\infty]$ is monotone decreasing and thus $\widehat{\lambda}_{\mathrm{RSIC}} = \infty$. (D2) $v_1 \neq v_2$, $(v_1-v_2)^2 \leq v_3$, and $v_1 = 0$ do not happen as shown in (C2).

(E) If $v_1 = v_2 = 0$, Eq.(64) yields $\mathrm{RSIC}[\lambda;\gamma] = 0$. Therefore, $\widehat{\lambda}_{\mathrm{RSIC}}$ is an arbitrary value in $[0,\infty)$.

(F) If $v_1 = v_2 > 0$, Eq.(64) implies that $\mathrm{RSIC}[\lambda;\gamma]$ does not depend on γ and is given by the right-hand side of Eq.(76). This implies that $\mathrm{RSIC}[\lambda;\gamma]$ is monotone decreasing with respect to λ and thus $\widehat{\lambda}_{\mathrm{RSIC}} = \infty$.

By summarizing the above results, we have Eq.(47). ■

References

[1] H. Akaike, "A new look at the statistical model identification," IEEE Transactions on Automatic Control, vol.AC-19, no.6, pp.716–723, 1974.

[2] A. Albert, Regression and the Moore-Penrose Pseudoinverse, Academic Press, New York and London, 1972.

[3] N. Aronszajn, "Theory of reproducing kernels," Transactions of the American Mathematical Society, vol.68, pp.337–404, 1950.

[4] C.M. Bishop, Neural Networks for Pattern Recognition, Clarendon Press, Oxford, 1995.

[5] F. Girosi, "An equivalence between sparse approximation and support vector machines," Neural Computation, vol.10, no.6, pp.1455–1480, 1998.

[6] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer, New York, 2001.

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

[8] A.E. Hoerl and R.W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," Technometrics, vol.12, no.3, pp.55–67, 1970.

[9] M. Ishiguro, Y. Sakamoto, and G. Kitagawa, "Bootstrapping log likelihood and EIC, an extension of AIC," Annals of the Institute of Statistical Mathematics, vol.49, pp.411–434, 1997.

[10] G.S. Kimeldorf and G. Wahba, "Some results on Tchebycheffian spline functions," Journal of Mathematical Analysis and Applications, vol.33, no.1, pp.82–95, 1971.

[11] S. Konishi and G. Kitagawa, "Generalized information criteria in model selection," Biometrika, vol.83, pp.875–890, 1996.

[12] A. Luntz and V. Brailovsky, "On estimation of characters obtained in statistical procedure of recognition," Technicheskaya Kibernetica, vol.3, 1969. in Russian.

[13] C.L. Mallows, "Some comments on C_P," Technometrics, vol.15, no.4, pp.661–675, 1973.

[14] T. Poggio and F. Girosi, "Networks for approximation and learning," Proceedings of the IEEE, vol.78, no.9, pp.1481–1497, 1990.

[15] B. Schölkopf and A.J. Smola, Learning with Kernels, MIT Press, Cambridge, MA, 2002.

[16] R. Shibata, "Statistical aspects of model selection," in From Data to Model, ed. J.C. Willems, pp.215–240, Springer-Verlag, New York, 1989.

[17] M. Sugiyama, "Active learning in approximately linear regression based on conditional expectation of generalization error," Journal of Machine Learning Research, vol.7, no.Jan, pp.141–166, 2006.

[18] M. Sugiyama, M. Kawanabe, and K.R. Müller, "Trading variance reduction with unbiasedness: The regularized subspace information criterion for robust model selection in kernel regression," Neural Computation, vol.16, no.5, pp.1077–1104, 2004.

[19] M. Sugiyama and K.R. Müller, "The subspace information criterion for infinite dimensional hypothesis spaces," Journal of Machine Learning Research, vol.3, no.Nov, pp.323–359, 2002.

[20] M. Sugiyama and H. Ogawa, "Subspace information criterion for model selection," Neural Computation, vol.13, no.8, pp.1863–1889, 2001.

[21] M. Sugiyama and H. Ogawa, "Optimal design of regularization term and regularization parameter by subspace information criterion," Neural Networks, vol.15, no.3, pp.349–361, 2002.

[22] A.N. Tikhonov and V.Y. Arsenin, Solutions of Ill-Posed Problems, V. H. Winston, Washington DC, 1977.

[23] V.N. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.

[24] G. Wahba, Spline Model for Observational Data, Society for Industrial and Applied Mathematics, Philadelphia and Pennsylvania, 1990.