IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol.E90-A, no.11, pp.2584-2592, 2007.

Analytic Optimization of Adaptive Ridge Parameters based on Regularized Subspace Information Criterion

Shun Gokita
Department of Computer Science
Tokyo Institute of Technology
2-12-1 O-okayama, Meguro-ku, Tokyo 152-8552, Japan

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
Department of Computational Intelligence and Systems Science
Tokyo Institute of Technology
4259 Nagatsuta-cho, Midori-ku, Yokohama 226-8502, Japan

Abstract

In order to obtain better learning results in supervised learning, it is important to choose model parameters appropriately. Model selection is usually carried out by preparing a finite set of model candidates, estimating a generalization error for each candidate, and choosing 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 focus on a generalization error estimator called the regularized subspace information criterion and derive an analytic form of the optimal model parameter over a set of infinitely many model candidates. This allows us to maximize the optimization quality while the computational cost is kept moderate.

Keywords

supervised learning, generalization error, model selection, regularized subspace information criterion

1 Introduction

The goal of supervised learning is to estimate an unknown function from training samples. If a good approximation of the target function is obtained, we can predict output values for unseen input points. This is called the generalization ability. The level of the generalization ability is usually measured by a distance between learned and target functions, which is referred to as the generalization error.

So far, various supervised learning methods have been developed, e.g., ridge learning [5]. Most of the learning methods contain model parameters such as the ridge parameter and the choice of the model parameters is crucial for better performance. Ideally, we want to choose the model parameters so that the generalization error is minimized. However, since the target function is unknown, we can not directly choose the model parameters as such. In practice, we use an estimator of the generalization error instead.

The regularized subspace information criterion (RSIC) [9] is a generalization error estimator for kernel models[1] and is shown to work well in practice. However, RSIC involves the optimization of additional tuning parameters, which is computationally rather expensive. In this paper, we derive an analytic form of the optimal model and tuning parameters over sets of infinitely many candidates. This maximally enhances the optimization quality, while the computational cost is kept moderate. Through simulations with artificial and benchmark data sets, we show that the new method tends to give comparable generalization performances with less computational costs.

2 Formulation of Supervised Learning

In this section, we briefly formulate the supervised learning problem (see the paper [12] for more detail).

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 $\mathcal{D} \subset \mathbb{R}^d$ and belongs to a reproducing kernel Hilbert space $\mathcal{H}$ [2]. We denote the reproducing kernel of $\mathcal{H}$ by $K(\boldsymbol{x}, \boldsymbol{x}')$. Let $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ be 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:

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

Let $\widehat{f}(\boldsymbol{x})$ be a learned function obtained from the training samples. The goal to obtain the best approximation to the target function in terms of the following generalization error.

$$\begin{aligned} G &= \|\widehat{f} - f\|^2 - \|f\|^2 \\ &= \|\widehat{f}\|^2 - 2\langle \widehat{f}, f \rangle, \end{aligned} \tag{2}$$

[1] This does not include multi-layer neural networks or radial basis function networks with adaptive centers.

where $\|f\|^2$ is subtracted for making the following discussion simple. We employ the following kernel model for learning [6].

$$\widehat{f}(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i K(\boldsymbol{x}, \boldsymbol{x}_i), \tag{3}$$

where $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_n)^\top$ are parameters to be learned and $^\top$ denotes the transpose. Note that this form is known to be a minimizer of some regularized functional in $\mathcal{H}$ [6]. We focus on learning the parameter vector $\boldsymbol{\alpha}$ in a linear fashion, i.e., $\boldsymbol{\alpha}$ is obtained by

$$\boldsymbol{\alpha} = \boldsymbol{L}\boldsymbol{y}, \tag{4}$$

where $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)^\top$ and $\boldsymbol{L}$ is an n-dimensional matrix which is independent of the noise $\{\epsilon_i\}_{i=1}^n$. 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 $\boldsymbol{L}$ such 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 the generalization error estimators called the *subspace information criterion* (SIC) [11, 10] and its extension the *regularized SIC* (RSIC) [9].

3.1 Subspace Information Criterion

Let $\mathcal{S}$ be the 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 $\mathcal{S}$. Since $g(\boldsymbol{x})$ belongs to $\mathcal{S}$, it is expressed as

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

where $\boldsymbol{\alpha}^* = (\alpha_1^*, \alpha_2^*, \ldots, \alpha_n^*)^\top$ are unknown optimal parameters. Then the expected generalization error G can be expressed as follows [10].

$$\mathbb{E}_{\boldsymbol{\epsilon}} G[\boldsymbol{L}] = \mathbb{E}_{\boldsymbol{\epsilon}} \langle \boldsymbol{K}\boldsymbol{L}\boldsymbol{y}, \boldsymbol{L}\boldsymbol{y} \rangle - 2\mathbb{E}_{\boldsymbol{\epsilon}} \langle \boldsymbol{K}\boldsymbol{L}\boldsymbol{y}, \boldsymbol{\alpha}^* \rangle, \tag{6}$$

where $\mathbb{E}_{\boldsymbol{\epsilon}}$ is the expectation over the noise $\{\epsilon\}_{i=1}^n$ and $K_{i,j} = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is the *kernel matrix*.

Since $\boldsymbol{\alpha}^*$ is unknown in Eq.(6), we replace it by a linear unbiased estimator $\widehat{\boldsymbol{\alpha}}_u$. Namely, with some n-dimensional matrix $\boldsymbol{R}_u$, $\widehat{\boldsymbol{\alpha}}_u$ is given as

$$\widehat{\boldsymbol{\alpha}}_u = \boldsymbol{R}_u \boldsymbol{y}, \tag{7}$$

which satisfies $\mathbb{E}_{\boldsymbol{\epsilon}}\widehat{\boldsymbol{\alpha}}_u = \boldsymbol{\alpha}^*$. Note that the subscript 'u' in the above equations stands for 'unbiased'. It is known that such $\boldsymbol{R}_u$ is given as follows [10].

$$\boldsymbol{R}_u = \boldsymbol{K}^\dagger, \tag{8}$$

where † denotes the Moore-Penrose generalized inverse [1].

Using $\widehat{\boldsymbol{\alpha}}_u$, we can express $\mathbb{E}_{\boldsymbol{\epsilon}}G$ as

$$\mathbb{E}_{\boldsymbol{\epsilon}}G[\boldsymbol{L}] = \mathbb{E}_{\boldsymbol{\epsilon}}\langle \boldsymbol{KLy}, \boldsymbol{Ly}\rangle - 2\mathbb{E}_{\boldsymbol{\epsilon}}\langle \boldsymbol{KLy}, \boldsymbol{R}_u\boldsymbol{y}\rangle + 2\sigma^2\mathrm{tr}(\boldsymbol{KLR}_u^\top). \tag{9}$$

The *subspace information criterion* (SIC) is defined as the right-hand side of Eq.(9) with the expectation operator $\mathbb{E}_{\boldsymbol{\epsilon}}$ removed:

$$\mathrm{SIC}[\boldsymbol{L}] = \langle \boldsymbol{KLy}, \boldsymbol{Ly}\rangle - 2\langle \boldsymbol{KLy}, \boldsymbol{R}_u\boldsymbol{y}\rangle + 2\sigma^2\mathrm{tr}(\boldsymbol{KLR}_u^\top). \tag{10}$$

For any $\boldsymbol{L}$, SIC is an unbiased estimator of $\mathbb{E}_{\boldsymbol{\epsilon}}G$:

$$\mathbb{E}_{\boldsymbol{\epsilon}}\mathrm{SIC}[\boldsymbol{L}] = \mathbb{E}_{\boldsymbol{\epsilon}}G[\boldsymbol{L}]. \tag{11}$$

The papers [11, 10] proposed choosing the learning matrix $\boldsymbol{L}$ that minimizes SIC from a set $\mathcal{L}$ of candidates of $\boldsymbol{L}$:

$$\widehat{\boldsymbol{L}} = \operatorname*{argmin}_{\boldsymbol{L}\in\mathcal{L}} \mathrm{SIC}[\boldsymbol{L}]. \tag{12}$$

3.2 Regularized Subspace Information Criterion

SIC is unbiased, but can have large variance. The *regularized SIC* (RSIC) can ease this problem [9].

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

$$\widehat{\boldsymbol{\alpha}}_r = \boldsymbol{Ry}, \tag{13}$$

where $\boldsymbol{R}$ is an n-dimensional matrix which is independent of the noise $\{\epsilon\}_{i=1}^n$. 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 the 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 $\widehat{\boldsymbol{\alpha}}_r$:

$$\mathrm{RSIC}[\boldsymbol{L};\boldsymbol{R}] = \langle \boldsymbol{KLy}, \boldsymbol{Ly}\rangle - 2\langle \boldsymbol{KLy}, \boldsymbol{Ry}\rangle + 2\sigma^2\mathrm{tr}(\boldsymbol{KLR}^\top), \tag{14}$$

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. The paper [9] proposed using the following criterion for optimizing $\boldsymbol{R}$:

$$J[\boldsymbol{R};\boldsymbol{L}] = (\mathrm{RSIC}[\boldsymbol{L};\boldsymbol{R}] - \mathbb{E}_{\boldsymbol{\epsilon}}G[\boldsymbol{L}])^2, \tag{15}$$

where the notation $J[\boldsymbol{R};\boldsymbol{L}]$ means that it is a functional of $\boldsymbol{R}$ with a parameter matrix $\boldsymbol{L}$. 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 $\boldsymbol{S}$ and $\boldsymbol{T}$ be

$$\boldsymbol{S} = 2\boldsymbol{R}_u^\top \boldsymbol{K}\boldsymbol{L} - 2\boldsymbol{R}^\top \boldsymbol{K}\boldsymbol{L}, \tag{16}$$

$$\boldsymbol{T} = \boldsymbol{L}^\top \boldsymbol{K}\boldsymbol{L} - 2\boldsymbol{R}^\top \boldsymbol{K}\boldsymbol{L}. \tag{17}$$

Then an unbiased estimator of $\mathbb{E}_{\boldsymbol{\epsilon}} J$ is given as follows [9].

$$\begin{aligned}\widehat{J}[\boldsymbol{R};\boldsymbol{L}] = &\left\{\langle \boldsymbol{S}\boldsymbol{y},\boldsymbol{y}\rangle - \sigma^2 \mathrm{tr}(\boldsymbol{S})\right\}^2 \\ &- \sigma^2\|(\boldsymbol{S}+\boldsymbol{S}^\top)\boldsymbol{y}\|^2 + \sigma^4 \mathrm{tr}(\boldsymbol{S}^2 + \boldsymbol{S}\boldsymbol{S}^\top) \\ &+ \sigma^2\|(\boldsymbol{T}+\boldsymbol{T}^\top)\boldsymbol{y}\|^2 - \sigma^4 \mathrm{tr}(\boldsymbol{T}^2 + \boldsymbol{T}\boldsymbol{T}^\top),\end{aligned} \tag{18}$$

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

$$\mathbb{E}_{\boldsymbol{\epsilon}} \widehat{J}[\boldsymbol{R};\boldsymbol{L}] = \mathbb{E}_{\boldsymbol{\epsilon}} J[\boldsymbol{R};\boldsymbol{L}]. \tag{19}$$

The paper [9] 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 $\mathcal{R}$:

$$\widehat{\boldsymbol{R}}^{(\boldsymbol{L})} = \operatorname*{argmin}_{\boldsymbol{R}\in\mathcal{R}} \widehat{J}[\boldsymbol{R};\boldsymbol{L}]. \tag{20}$$

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

$$\widehat{\boldsymbol{L}} = \operatorname*{argmin}_{\boldsymbol{L}\in\mathcal{L}} \mathrm{RSIC}[\boldsymbol{L};\widehat{\boldsymbol{R}}^{(\boldsymbol{L})}]. \tag{21}$$

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 the set $\mathcal{R}$ from which $\boldsymbol{R}$ is searched. 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 [5] determines the parameter $\boldsymbol{\alpha}$ so that the regularized squared error is minimized.

$$\sum_{i=1}^{n} \left(\widehat{f}(\boldsymbol{x}_i) - y_i\right)^2 + \eta\|\boldsymbol{\alpha}\|^2, \tag{22}$$

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

$$\boldsymbol{L} = (\boldsymbol{K}^2 + \eta \boldsymbol{I})^{-1} \boldsymbol{K}, \tag{23}$$

where $\boldsymbol{I}$ is the identity matrix. When RSIC is employed, we have to optimize the reference matrix $\boldsymbol{R}$ in addition to the learning matrix $\boldsymbol{L}$. Here, we use ridge learning also for $\boldsymbol{R}$:

$$\boldsymbol{R} = (\boldsymbol{K}^2 + \nu \boldsymbol{I})^{-1} \boldsymbol{K}, \tag{24}$$

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

The paper [9] 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 E1. The computational complexity of E1 with respect to the number of model candidates is $\mathcal{O}(|\mathcal{L}||\mathcal{R}|)$, where $|\mathcal{L}|$ and $|\mathcal{R}|$ are the size of the sets $\mathcal{L}$ and $\mathcal{R}$ respectively. The procedure E1 has been shown to work well, given that the sets $\mathcal{L}$ and $\mathcal{R}$ are rich enough [9].

4.2 Existing Method 2 (E2)

In the procedure E1, $\boldsymbol{R}$ and $\boldsymbol{L}$ are chosen from a finite set of candidates. In order to improve the optimization quality of $\boldsymbol{R}$ and $\boldsymbol{L}$, it is desirable to increase the number of candidates. However, this in turn increases the computational complexity. To cope with this problem, the paper [12] proposed an efficient model selection procedure based on RSIC, 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 cost reasonable.

It appears to be difficult to have an analytic solution if the set $\mathcal{R}$ and $\mathcal{L}$ are determined based on ridge learning (23) and (24) since the target parameters η and ν are included in the matrix inverse. The paper [12] instead employed *shrinkage learning*: determine the parameter $\boldsymbol{\alpha}$ so that the following quantity is minimized.

$$\sum_{i=1}^{n} \left(\widehat{f}(\boldsymbol{x}_i) - y_i \right)^2 + \delta \|\boldsymbol{K}\boldsymbol{\alpha}\|^2, \tag{25}$$

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

$$\boldsymbol{L} = \frac{1}{1+\delta} \boldsymbol{K}^{\dagger}. \tag{26}$$

We also employ shrinkage learning for $\boldsymbol{R}$:

$$\boldsymbol{R} = \frac{1}{1+\kappa} \boldsymbol{K}^{\dagger}, \tag{27}$$

where κ is a non-negative scalar. Now the problem of choosing $\boldsymbol{R}$ and $\boldsymbol{L}$ is reduced to the problem of choosing κ and δ. Let $\widehat{\kappa}^{(\delta)}$ and $\widehat{\delta}$ be

$$\widehat{\kappa}^{(\delta)} = \operatorname*{arginf}_{\kappa \in [0,\infty)} \widehat{J}(\kappa; \delta), \tag{28}$$

$$\widehat{\delta} = \operatorname*{arginf}_{\delta \in [0,\infty)} \mathrm{RSIC}(\delta; \widehat{\kappa}^{(\delta)}), \tag{29}$$

and let

$$v_1 = \langle \boldsymbol{K}^\dagger \boldsymbol{y}, \boldsymbol{y} \rangle, \tag{30}$$

$$v_2 = \sigma^2 \mathrm{tr}(\boldsymbol{K}^\dagger), \tag{31}$$

$$v_3 = 2\sigma^2 \langle (\boldsymbol{K}^\dagger)^2 \boldsymbol{y}, \boldsymbol{y} \rangle - \sigma^4 \mathrm{tr}((\boldsymbol{K}^\dagger)^2). \tag{32}$$

Then $\widehat{\delta}$ is given as follows [12].

$$\widehat{\delta} = \begin{cases} \dfrac{(v_1 - v_2)v_2}{(v_1 - v_2)^2 - 2\max(0, v_3)} & \text{if } v_1 > v_2 \text{ and } v_3 < \dfrac{(v_1 - v_2)^2}{2}, \\ \text{arbitrary value in } [0, \infty) & \text{if } v_1 = v_2 = 0, \\ \infty & \text{otherwise.} \end{cases} \tag{33}$$

By this expression, we can compute the optimal value of δ analytically. We refer to this procedure as E2. The computational complexity of E2 is $\mathcal{O}(1)$ with respect to $|\mathcal{L}|$ and $|\mathcal{R}|$.

5 Proposed Method for Determining $\boldsymbol{L}$

In the previous section, we reviewed existing RSIC-based methods of determining $\boldsymbol{L}$. As we experimentally show in Section 6, E1 works excellently in terms of the generalization error, but it is computationally less efficient. On the other hand, E2 is computationally very efficient, but it works poorly in terms of the generalization error. A reason for the poor performance of E2 is shrinkage learning—its regularization effect is rather limited. Although ridge learning works better, it is hard to derive an analytic form of the optimizer since the ridge parameter is included in the matrix inverse. In this section, we propose a new learning method that is more powerful than ridge learning, but it still allows us to obtain an analytic form of the optimizer.

5.1 Adaptive Ridge Learning in Kernel Eigenspace

Let $d_1, d_2, \ldots, d_n$ be the eigenvalues of $\boldsymbol{K}$ and let $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_n$ be the associated normalized eigenvectors. Let

$$\boldsymbol{D} = \mathrm{diag}(d_1, d_2, \ldots, d_n), \tag{34}$$

$$\boldsymbol{U} = (\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_n), \tag{35}$$

$$\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_n)^\top = \boldsymbol{U}^\top \boldsymbol{\alpha}, \tag{36}$$

where $\mathrm{diag}(d_1, d_2, \ldots, d_n)$ denotes the diagonal matrix with diagonal elements $d_1, d_2, \ldots, d_n$. *Adaptive ridge learning in kernel eigenspace* determines the parameter $\boldsymbol{\alpha}$ so that the following criterion is minimized.

$$\sum_{i=1}^{n}(\hat{f}(x_i) - y_i)^2 + \sum_{j=1}^{n}\lambda_j\beta_j, \tag{37}$$

where $\{\lambda_j\}_{j=1}^n$ are non-negative scalars. Adaptive ridge learning in kernel eigenspace contains n independent ridge parameters $\{\lambda_j\}_{j=1}^n$. If $\lambda_1 = \lambda_2 = \cdots = \lambda_n = \lambda$, adaptive ridge learning in kernel eigenspace is reduced to ordinary ridge learning; if $\lambda_i = \lambda d_i^2$, it agrees with shrinkage learning. Thus, adaptive ridge learning in kernel eigenspace includes ridge learning and shrinkage learning as special cases and is substantially more general. Let

$$\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n). \tag{38}$$

Then a minimizer of Eq.(37) is given by

$$\boldsymbol{L} = (\boldsymbol{K}^2 + \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^\top)^\dagger\boldsymbol{K}. \tag{39}$$

5.2 Proposed Method (P)

Here, we derive an analytic form of the optimal $\{\lambda_i\}_{i=1}^n$ that minimize RSIC for any fixed $\boldsymbol{R}$.

Let $\boldsymbol{s}$, $\boldsymbol{t}$, $\boldsymbol{W}$, and p_i be

$$\boldsymbol{s} = (s_1, s_2, \ldots, s_n)^\top = \boldsymbol{U}^\top\boldsymbol{y}, \tag{40}$$
$$\boldsymbol{t} = (t_1, t_2, \ldots, t_n)^\top = \boldsymbol{U}^\top\boldsymbol{R}\boldsymbol{y}, \tag{41}$$
$$\boldsymbol{W} = (W_{i,j}) = \boldsymbol{U}^\top\boldsymbol{R}^\top\boldsymbol{U}, \tag{42}$$
$$p_i = \sigma^2 W_{i,i} - s_i t_i. \tag{43}$$

Then we have the following theorem.

Theorem 1 *Let*

$$(\widehat{\lambda}_1^{(\boldsymbol{R})}, \widehat{\lambda}_2^{(\boldsymbol{R})}, \ldots, \widehat{\lambda}_n^{(\boldsymbol{R})}) = \operatorname*{arginf}_{(\lambda_1,\lambda_2,\ldots,\lambda_n)\in[0,\infty)^n} \mathrm{RSIC}(\lambda_1, \lambda_2, \ldots, \lambda_n; \boldsymbol{R}). \tag{44}$$

Then $\widehat{\lambda}_i^{(\boldsymbol{R})}$ *is given by*

$$\widehat{\lambda}_i^{(\boldsymbol{R})} = \begin{cases} \max\left(0, -\dfrac{d_i s_i^2}{p_i} - d_i^2\right) & \text{if } d_i > 0 \text{ and } p_i < 0, \quad (45a) \\ \text{arbitrary value in } [0,\infty) & \text{if } d_i = 0, \quad (45b) \\ \text{arbitrary value in } [0,\infty) & \text{if } d_i > 0 \text{ and } s_i = p_i = 0, \quad (45c) \\ \infty & \text{otherwise.} \quad (45d) \end{cases}$$

A proof of this theorem is given in A. By Theorem 1, the optimal value of $\{\lambda_i\}_{i=1}^n$ can be analytically calculated for any fixed $\boldsymbol{R}$.

To further optimize $\boldsymbol{R}$, we employ shrinkage learning for $\boldsymbol{R}$:

$$\boldsymbol{R} = \frac{1}{1+\gamma}\boldsymbol{K}^\dagger, \tag{46}$$

where γ is a non-negative scalar. Then we have the following corollary.

Corollary 1 *When Eq.(46) is used for $\boldsymbol{R}$, Eq.(45) can be expressed as*

$$\widehat{\lambda}_i^{(\gamma)} = \begin{cases} \dfrac{d_i^2(\gamma s_i^2 + \sigma^2)}{s_i^2 - \sigma^2} & \text{if } d_i > 0 \text{ and } s_i^2 > \sigma^2, \quad (47a) \\ \text{arbitrary value in } [0,\infty) & \text{if } d_i = 0, \quad (47b) \\ \text{arbitrary value in } [0,\infty) & \text{if } d_i > 0 \text{ and } s_i = \sigma = 0, \quad (47c) \\ \infty & \text{otherwise.} \quad (47d) \end{cases}$$

A proof of this corollary is given in B.

Now we derive an analytic form of the optimal γ. Let

$$\widehat{\Lambda}^{(\gamma)} = \mathrm{diag}(\widehat{\lambda}_1^{(\gamma)}, \widehat{\lambda}_2^{(\gamma)}, \cdots, \widehat{\lambda}_n^{(\gamma)}), \tag{48}$$

$$\widehat{\boldsymbol{L}}^{(\gamma)} = (\boldsymbol{K}^2 + \boldsymbol{U}\widehat{\boldsymbol{\Lambda}}^{(\gamma)}\boldsymbol{U}^\top)^\dagger \boldsymbol{K}. \tag{49}$$

$\widehat{J}$ defined by Eq.(18) is a criterion which measures the goodness of γ for a fixed $\boldsymbol{L}$. However, $\widehat{\boldsymbol{L}}^{(\gamma)}$ given above depends on γ. Therefore, determining γ by $\widehat{J}(\gamma; \widehat{\boldsymbol{L}}^{(\gamma)})$ may not be appropriate. Here, we propose using the following $\widehat{J}_E(\gamma)$ for measuring the goodness of γ.

$$\widehat{J}_E(\gamma) = \int_0^\infty \widehat{J}(\gamma; \widehat{\boldsymbol{L}}^{(\gamma')}) d\gamma', \tag{50}$$

which is the average of $\widehat{J}$ over $\widehat{\boldsymbol{L}}^{(\gamma')}$.

Let $A_{i,j}$, B_i, A, and B be

$$A_{i,j} = \begin{cases} \dfrac{(s_i^2 - \sigma^2)^2 (s_j^2 - \sigma^2)^2}{d_i d_j s_i^2 s_j^2} & \text{if } d_i, d_j > 0 \text{ and } s_i^2, s_j^2 > \sigma^2, \\ 0 & \text{otherwise,} \end{cases} \tag{51}$$

$$B_i = \begin{cases} \dfrac{\sigma^2(3s_i^2 + \sigma^2)(s_i^2 - \sigma^2)^2(2s_i^2 - \sigma^2)}{2d_i^2 s_i^6} & \text{if } d_i > 0 \text{ and } s_i^2 > \sigma^2, \\ 0 & \text{otherwise,} \end{cases} \tag{52}$$

$$A = \sum_{i=1}^n \sum_{j=1}^n A_{i,j}, \tag{53}$$

$$B = \sum_{i=1}^n B_i. \tag{54}$$

Then we have the following theorem.

Theorem 2 *Let*

$$\widehat{\gamma} = \underset{\gamma \in [0,\infty)}{\operatorname{arginf}} \widehat{J}_E(\gamma). \tag{55}$$

Then $\widehat{\gamma}$ *is given by*

$$\widehat{\gamma} = \begin{cases} \max\left(0, \dfrac{B}{A-B}\right) & \text{if } A - B > 0, \\ \text{arbitrary value in } [0,\infty) & \text{if } A = B = 0, \\ \infty & \text{otherwise.} \end{cases} \tag{56}$$

A proof of this theorem is given in C. By Eqs.(47) and (56), we can analytically compute the optimal ridge parameters $\{\widehat{\lambda}_i^{(\widehat{\gamma})}\}_{i=1}^n$. We refer to this procedure as P. The computational complexity of P is $\mathcal{O}(1)$ with respect to $|\mathcal{L}|$ and $|\mathcal{R}|$.

6 Simulations

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

6.1 Toy Data Set

First, we illustrate how the proposed method works using a simple artificial simulation.

Let $f(x) = \mathrm{sinc}(x)$ and we employ the Gaussian reproducing kernel Hilbert space [8] as $\mathcal{H}$, where the reproducing kernel is given by

$$K(x, x') = \exp\left(-\frac{(x-x')^2}{2}\right). \tag{57}$$

Note that the sinc function is included in the above Gaussian reproducing kernel Hilbert space [3]. 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.

$$\begin{aligned}(n, \sigma^2) =& (50, 0.01), (50, 0.09), \\ & (100, 0.01), (100, 0.09).\end{aligned} \tag{58}$$

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 unknown variable and is estimated by[2]

$$\hat{\sigma^2} = \frac{\|\boldsymbol{K}\boldsymbol{K}^\dagger \boldsymbol{y} - \boldsymbol{y}\|^2}{n - \mathrm{tr}(\boldsymbol{K}\boldsymbol{K}^\dagger)}. \tag{59}$$

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

$$\bar{G} = \|\hat{f} - f\|^2. \tag{60}$$

η and ν in the method E1 are chosen from the set of 10 equidistant values in log-scale in the range $[10^{-4}, 10^4]$. Therefore, $|\mathcal{L}| = |\mathcal{R}| = 10$. 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. The computation of the Moore-Penrose generalized inverse was often numerically unstable, so we discarded eigenvalues less than 0.02.

Mean and standard deviation of the generalization error obtained by each method are described in the upper area of Table 1. The ratio of mean generalization error among the methods is also described in the table for better comparison. '+' ('−') signifies that P gives a significantly better (worse) result by the *t-test* [4] at the significance level 1%. The table also contains the ratio of mean computation time among the methods.

The computational complexity of E1 is $\mathcal{O}(|\mathcal{L}||\mathcal{R}|)$ while that of P is $\mathcal{O}(1)$. Thus P is theoretically 100 times faster than E1 in computation. Table 1 shows that practical computation time of P is approximately 20 times faster than E1, which is rather consistent with the theoretical value. The generalization error of P is comparable to that of E1.

The computational complexity of P and E2 are both $\mathcal{O}(1)$. Table 1 shows that although they have the same computational complexity, P required a few times more computation time than E2. We conjecture that this is mainly due to the computation of Eqs.(47) and (56). The generalization error of P is better than E2 in all case. Especially, when the noise level is high ($\sigma^2 = 0.09$), the generalization error is highly improved.

The above results show that, in this illustrative simulations, P is much faster than E1 while the generalization performance is comparable; compared with E2, P is a few times slower but the generalization performance is much better.

6.2 Benchmark Data Sets

Here, we apply the proposed method to more realistic data sets and evaluate their behavior. We use 10 benchmark data sets provided by DELVE [7]: *Abalone, Boston, Bank-8fm, Bank-8nm, Bank-8fh, Bank-8nh, Kin-8fm, Kin-8nm, Kin-8fh*, and *Kin-8nh*.

For convenience, every attribute is normalized to [0,1]. n randomly selected samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ are used for training. We evaluate the generalization performance by the

[2]Note that the current setting theoretically yields $\boldsymbol{K}\boldsymbol{K}^\dagger = \boldsymbol{I}$ (i.e., $\boldsymbol{K}^{-1}$ exists) with probability one [8]. However, in practice, $\boldsymbol{K}$ is almost always degenerated numerically and therefore Eq.(59) is still valid.

mean squared test error:

$$\text{Test Error} = \frac{1}{n'} \sum_{i=1}^{n'} \left(\hat{f}(\boldsymbol{x}'_i) - y'_i \right)^2, \tag{61}$$

where $\{(\boldsymbol{x}'_i, y'_i)\}_{i=1}^{n'}$ denote the test samples which are not used for training. Other setting is the same as Section 6.1. The simulation is repeated 100 times, randomly selecting the training set $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$ in each trial. We test $n = 100$ and 800 (400 for the Boston data set since it contains only 506 samples).

Mean and standard deviation of the generalization error obtained by each method are described in the lower half of Table 1. The table shows that, irrespective of n, E1 gives the best performance for most of the data sets; P is slightly inferior to E1 in terms of the generalization performance, but is much better than E2. When $n = 100$, the computation time of P is approximately 20 times faster than E1 and is a few times slower than E2. However, when $n = 800$, the computation time of P gets relatively slow. Therefore, P would be particularly useful in small sample cases.

7 Conclusions

In this paper, we proposed a new learning method called adaptive ridge learning in kernel eigenspace and derived an analytic form of the optimal ridge parameters. We experimentally showed that for some toy and benchmark data sets, the proposed method is computationally more efficient than the existing method with grid search, while the generalization performance is kept rather comparable. In particular, our experiments highlighted that the proposed method is useful when the number of training samples is small.

Acknowledgements

This work was supported by MEXT Grant-in-Aid for Young Scientists 17700142 and Grant-in-Aid for Scientific Research (B) 18300057.

A Proof of Theorem 1

By Eqs.(14), and (39), $\mathrm{RSIC}(\lambda_1, \lambda_2, \ldots, \lambda_n; \boldsymbol{R})$ is expressed as

$$\begin{aligned}
&\mathrm{RSIC}(\lambda_1, \lambda_2, \ldots, \lambda_n; \boldsymbol{R}) = \\
&\quad \langle \{(\boldsymbol{D}^2 + \boldsymbol{\Lambda})^\dagger\}^2 \boldsymbol{D}^3 \boldsymbol{s}, \boldsymbol{s} \rangle - 2 \langle (\boldsymbol{D}^2 + \boldsymbol{\Lambda})^\dagger \boldsymbol{D}^2 \boldsymbol{s}, \boldsymbol{t} \rangle \\
&\quad + 2\sigma^2 \mathrm{tr} \left(\boldsymbol{W} (\boldsymbol{D}^2 + \boldsymbol{\Lambda})^\dagger \boldsymbol{D}^2 \right) \\
&\quad = \sum_{i=1}^{n} h_i(\lambda_i),
\end{aligned} \tag{62}$$

Table 1: Simulation results. '+' ('−') means that P gives a significantly better (worse) result by the t-test at the significance level 1%. Generalization errors of the DELVE data sets are multiplied by 10^3.

	Mean and Std. of Gen. Error			Ratio of Mean Gen. Error		Ratio of Mean Comp. Time	
Data	P	E1	E2	P/E1	P/E2	P/E1	P/E2
Toy ($n = 50, \sigma^2 = 0.01$)	0.93(0.27)	0.91(0.26)	1.03(0.43)	1.02	0.90^+	0.05	2.18
Toy ($n = 50, \sigma^2 = 0.09$)	2.28(1.59)	2.17(1.92)	3.92(2.68)	1.05	0.58^+	0.04	2.10
Toy ($n = 100, \sigma^2 = 0.01$)	0.81(0.19)	0.84(0.13)	0.84(0.30)	0.96^+	0.96^+	0.06	1.92
Toy ($n = 100, \sigma^2 = 0.09$)	1.51(1.07)	1.51(1.59)	2.81(1.97)	1.00	0.54^+	0.05	2.66
Abalone ($n = 100$)	8.45(2.35)	6.83(0.43)	13.20(7.30)	1.24^-	0.64^+	0.05	1.40
Boston ($n = 100$)	11.10(2.22)	10.95(2.33)	20.80(8.36)	1.01	0.53^+	0.05	1.17
Bank-8fm ($n = 100$)	3.40(0.35)	3.40(0.45)	5.44(1.10)	1.00	0.62^+	0.04	2.00
Bank-8nm ($n = 100$)	5.49(0.62)	5.16(0.55)	6.91(1.28)	1.06^-	0.79^+	0.06	3.50
Bank-8fh ($n = 100$)	12.20(1.16)	11.85(1.90)	33.33(13.42)	1.03	0.37^+	0.03	1.33
Bank-8nh ($n = 100$)	17.44(2.75)	16.68(1.58)	28.60(6.64)	1.05	0.61^+	0.07	4.50
Kin-8fm ($n = 100$)	1.69(0.25)	1.72(0.31)	14.00(6.19)	0.98	0.12^+	0.01	0.50
Kin-8nm ($n = 100$)	21.28(2.01)	20.35(1.83)	73.33(40.19)	1.05^-	0.29^+	0.06	1.25
Kin-8fh ($n = 100$)	9.16(1.31)	7.71(0.60)	161.57(94.65)	1.19^-	0.06^+	0.04	1.50
Kin-8nh ($n = 100$)	26.69(3.17)	24.24(2.26)	121.24(55.68)	1.10^-	0.22^+	0.05	1.33
Abalone ($n = 800$)	6.14(0.19)	6.00(0.13)	8.48(1.63)	1.02^-	0.72^+	0.57	7.40
Boston ($n = 400$)	5.68(1.70)	5.92(2.25)	10.38(3.29)	0.96	0.55^+	0.32	3.73
Bank-8fm ($n = 800$)	2.10(0.06)	2.09(0.05)	4.94(0.70)	1.01	0.43^+	0.58	5.43
Bank-8nm ($n = 800$)	2.76(0.10)	2.72(0.10)	3.21(0.28)	1.02^-	0.86^+	0.56	5.89
Bank-8fh ($n = 800$)	8.97(0.16)	8.72(0.17)	35.58(8.34)	1.03^-	0.25^+	0.56	6.94
Bank-8nh ($n = 800$)	13.23(0.29)	12.89(0.36)	23.40(3.83)	1.03^-	0.57^+	0.56	6.53
Kin-8fm ($n = 800$)	0.72(0.02)	0.71(0.02)	15.91(2.06)	1.02^-	0.05^+	0.52	4.42
Kin-8nm ($n = 800$)	10.29(0.44)	10.13(0.30)	21.41(4.52)	1.02^-	0.48^+	0.55	8.00
Kin-8fh ($n = 800$)	5.53(0.17)	5.11(0.09)	196.61(51.35)	1.08^-	0.03^+	0.57	8.21
Kin-8nh ($n = 800$)	16.44(0.48)	15.99(0.40)	56.20(11.08)	1.03^-	0.29^+	0.59	7.44

Table 2: Cases in proof of Theorem1.

Conditions			Results
$d_i = 0$			(A) $\widehat{\lambda}_i^{(\boldsymbol{R})} \in [0, \infty)$
$d_i > 0$	$s_i = 0$	$p_i > 0$	(B) $\widehat{\lambda}_i^{(\boldsymbol{R})} = \infty$
		$p_i = 0$	(C) $\widehat{\lambda}_i^{(\boldsymbol{R})} \in [0, \infty)$
		$p_i < 0$	(D) $\widehat{\lambda}_i^{(\boldsymbol{R})} = 0$
	$s_i \neq 0$	$p_i \geq 0$	(E) $\widehat{\lambda}_i^{(\boldsymbol{R})} = \infty$
		$p_i < 0$	(F) $\widehat{\lambda}_i^{(\boldsymbol{R})} = \max(0, \tilde{\lambda}_i)$

where

$$h_i(\lambda) = d_i^3 s_i^2 \{(d_i^2 + \lambda)^\dagger\}^2 + 2 d_i^2 p_i (d_i^2 + \lambda)^\dagger. \tag{63}$$

This implies that the minimizer of $\mathrm{RSIC}(\lambda_1, \lambda_2, \ldots, \lambda_n; \boldsymbol{R})$ with respect to λ_i is the minimizer of $h_i(\lambda)$. When $d_i \neq 0$, the first derivative of $h_i(\lambda)$ is given by

$$h_i'(\lambda) = -2 d_i^2 \frac{d_i(s_i^2 + d_i p_i) + p_i \lambda}{(d_i^2 + \lambda)^3}. \tag{64}$$

Below, we give a proof depending on d_i, s_i and p_i (see Table 2).

(A) If $d_i = 0$, $h_i(\lambda) = 0$ for any $\lambda \in [0, \infty)$. So $\widehat{\lambda}_i^{(\boldsymbol{R})}$ is an arbitrary value in $[0, \infty)$.

(B) If $d_i > 0$, $s_i = 0$ and $p_i > 0$, Eq.(64) yields $h_i'(\lambda) < 0$ for any $\lambda \in [0, \infty)$. This implies that $h_i(\lambda)$ is monotone decreasing and thus $\widehat{\lambda}_i^{(\boldsymbol{R})} = \infty$.

(C) If $d_i > 0$, $s_i = 0$ and $p_i = 0$, Eq.(64) yields $h_i'(\lambda) = 0$ for any $\lambda \in [0, \infty)$. This implies that $h_i(\lambda)$ is constant so $\widehat{\lambda}_i^{(\boldsymbol{R})}$ is an arbitrary value in $[0, \infty)$.

(D) If $d_i > 0$, $s_i = 0$ and $p_i < 0$, Eq.(64) yields $h_i'(\lambda) > 0$ for any $\lambda \in [0, \infty)$. This implies that $h_i(\lambda)$ is monotone increasing and thus $\widehat{\lambda}_i^{(\boldsymbol{R})} = 0$.

(E) If $d_i > 0$, $s_i \neq 0$ and $p_i \geq 0$, Eq.(64) yields $h_i'(\lambda) < 0$ for any $\lambda \in [0, \infty)$. This implies that $h_i(\lambda)$ is monotone decreasing and thus $\widehat{\lambda}_i^{(\boldsymbol{R})} = \infty$.

(F) If $d_i > 0$, $s_i \neq 0$, $p_i < 0$ and $s_i^2 + d_i p_i < 0$, Eq(64) yields $h_i'(\lambda) > 0$ for any $\lambda \in [0, \infty)$. This implies that $h_i(\lambda)$ is monotone increasing and thus $\widehat{\lambda}_i^{(\boldsymbol{R})} = 0$. On the other hand, if $s_i^2 + d_i p_i \geq 0$, Eq(64) implies that $h_i'(\tilde{\lambda}_i) = 0$, where

$$\tilde{\lambda}_i = -\frac{d_i(s_i^2 + d_i p_i)}{p_i} (\geq 0). \tag{65}$$

Since

$$h_i(\lambda) - h_i(\tilde{\lambda}_i) = \frac{d_i p_i^2 (\lambda - \tilde{\lambda}_i)^2}{s_i^2 (d_i^2 + \lambda)^2} \geq 0, \tag{66}$$

where strict equality holds if and only if $\lambda_i = \tilde{\lambda}_i$, we have $\widehat{\lambda}_i^{(\boldsymbol{R})} = \tilde{\lambda}_i$. Then we can express $\widehat{\lambda}_i^{(\boldsymbol{R})}$ as

$$\widehat{\lambda}_i^{(\boldsymbol{R})} = \max(0, \tilde{\lambda}_i). \tag{67}$$

By summarizing the above results (see Table 2), we have Eq.(45). ■

B Proof of Corollary 1

When $\boldsymbol{R}$ is of the form of Eq.(46), p_i is expressed as

$$p_i = \frac{d_i^\dagger(\sigma^2 - s_i^2)}{(1+\gamma)}. \tag{68}$$

Since $p_i < 0$ implies $s_i^2 > \sigma^2$, Eq.(45a) is expressed as

$$\max\left(0, -\frac{d_i s_i^2}{p_i} - d_i^2\right) = \frac{d_i^2(\gamma s_i^2 + \sigma^2)}{s_i^2 - \sigma^2} > 0. \tag{69}$$

$s_i = p_i = 0$ implies $s_i = \sigma = 0$, which concludes the proof. ■

C Proof of Theorem 2

Let

$$\boldsymbol{Q}_1 = \boldsymbol{K}^\dagger \boldsymbol{K}\boldsymbol{L}, \tag{70}$$
$$\boldsymbol{Q}_2 = \boldsymbol{L}^\top \boldsymbol{K}\boldsymbol{L}, \tag{71}$$
$$q_1 = \langle \boldsymbol{Q}_1 \boldsymbol{y}, \boldsymbol{y}\rangle - \sigma^2 \mathrm{tr}\left(\boldsymbol{Q}_1\right), \tag{72}$$
$$q_2 = \sigma^2\left\{\|(\boldsymbol{Q}_1 + \boldsymbol{Q}_1^\top)\boldsymbol{y}\|^2 - \sigma^2 \mathrm{tr}\left(\boldsymbol{Q}_1^2 + \boldsymbol{Q}_1^\top \boldsymbol{Q}_1\right)\right\}, \tag{73}$$
$$q_3 = \sigma^2\left\{\langle(\boldsymbol{Q}_1 + \boldsymbol{Q}_1^\top)\boldsymbol{Q}_2\boldsymbol{y}, \boldsymbol{y}\rangle - \sigma^2 \mathrm{tr}\left(\boldsymbol{Q}_1\boldsymbol{Q}_2\right)\right\}. \tag{74}$$

When $\boldsymbol{R}$ is of the form of Eq.(46), Eqs.(16) and (17) are expressed as

$$\boldsymbol{S} = \frac{2\gamma}{1+\gamma}\boldsymbol{Q}_1, \tag{75}$$
$$\boldsymbol{T} = \boldsymbol{Q}_2 - \frac{2}{1+\gamma}\boldsymbol{Q}_1. \tag{76}$$

Using Eqs.(72)—(76), we can express Eq.(18) as

$$\begin{aligned}\widehat{J}(\gamma; \boldsymbol{L}) =& 4(q_1^2 - q_2)\left(\frac{\gamma}{1+\gamma}\right)^2 + q_2\frac{4}{(1+\gamma)^2} \\ &- q_3\frac{8}{1+\gamma} + \sigma^2\{4\|\boldsymbol{Q}_2\boldsymbol{y}\|^2 - 2\sigma^2 \mathrm{tr}\left(\boldsymbol{Q}_2^2\right)\}.\end{aligned} \tag{77}$$

When $\boldsymbol{L}$ is of the form of Eq.(49), q_1, q_2 and q_3 depend on γ. So we denote them as $q_1^{(\gamma)}$, $q_2^{(\gamma)}$ and $q_3^{(\gamma)}$, which are expressed as

$$\begin{aligned} q_1^{(\gamma)} &= \langle (\boldsymbol{D}^2 + \widehat{\boldsymbol{\Lambda}}^{(\gamma)})^\dagger \boldsymbol{D}\boldsymbol{s}, \boldsymbol{s}\rangle - \sigma^2 \mathrm{tr}\left((\boldsymbol{D}^2 + \widehat{\boldsymbol{\Lambda}}^{(\gamma)})^\dagger \boldsymbol{D}\right) \\ &= \sum_{i=1}^{n} r_{1,i}^{(\gamma)}, \end{aligned} \tag{78}$$

$$\begin{aligned} q_2^{(\gamma)} &= \sigma^2 \left\{ 4\|\boldsymbol{U}(\boldsymbol{D}^2 + \widehat{\boldsymbol{\Lambda}}^{(\gamma)})^\dagger \boldsymbol{D}\boldsymbol{s}\|^2 - 2\sigma^2 \mathrm{tr}\left(\{(\boldsymbol{D}^2 + \widehat{\boldsymbol{\Lambda}}^{(\gamma)})^\dagger\}^2 \boldsymbol{D}^2\right)\right\} \\ &= \sum_{i=1}^{n} r_{2,i}^{(\gamma)}, \end{aligned} \tag{79}$$

$$\begin{aligned} q_3^{(\gamma)} &= \sigma^2 \left\{ 2\langle \{(\boldsymbol{D}^2 + \widehat{\boldsymbol{\Lambda}}^{(\gamma)})^\dagger\}^3 \boldsymbol{D}^4 \boldsymbol{s}, \boldsymbol{s}\rangle - \sigma^2 \mathrm{tr}\left(\{(\boldsymbol{D}^2 + \widehat{\boldsymbol{\Lambda}}^{(\gamma)})^\dagger\}^3 \boldsymbol{D}^4\right)\right\} \\ &= \sum_{i=1}^{n} r_{3,i}^{(\gamma)}, \end{aligned} \tag{80}$$

where

$$r_{1,i}^{(\gamma)} = d_i(s_i^2 - \sigma^2)(d_i^2 + \widehat{\lambda}_i^{(\gamma)})^\dagger, \tag{81}$$

$$r_{2,i}^{(\gamma)} = 2\sigma^2 d_i^2 (2s_i^2 - \sigma^2)\{(d_i^2 + \widehat{\lambda}_i^{(\gamma)})^\dagger\}^2, \tag{82}$$

$$r_{3,i}^{(\gamma)} = \sigma^2 d_i^4 (2s_i^2 - \sigma^2)\{(d_i^2 + \widehat{\lambda}_i^{(\gamma)})^\dagger\}^3. \tag{83}$$

Let A, B and C be

$$A = \int_0^\infty (q_1^{(\gamma')})^2 d\gamma', \tag{84}$$

$$B = \int_0^\infty (q_2^{(\gamma')} - q_3^{(\gamma')}) d\gamma', \tag{85}$$

$$C = \int_0^\infty q_3^{(\gamma')} d\gamma'. \tag{86}$$

Then Eq.(50) is expressed as

$$\begin{aligned} \widehat{J}_E(\gamma) =& 4\left(\frac{\gamma}{1+\gamma}\right)^2 (A - B - C) + \frac{4}{(1+\gamma)^2}(B + C) \\ & - \frac{8}{1+\gamma} C + \text{const.}, \end{aligned} \tag{87}$$

where "const." denotes a constant that does not depend on γ. The first derivative of $\widehat{J}_E(\gamma)$ is given by

$$\widehat{J}_E'(\gamma) = \frac{8\{(A-B)\gamma - B\}}{(1+\gamma)^3}. \tag{88}$$

Let $A_{i,j}$ and B_i be

$$A_{i,j} = \int_0^\infty r_{1,i}^{(\gamma')} r_{1,j}^{(\gamma')} d\gamma', \tag{89}$$

$$B_i = \int_0^\infty (r_{2,i}^{(\gamma')} - r_{3,i}^{(\gamma')}) d\gamma'. \tag{90}$$

When $\widehat{\lambda}_i^{(\gamma')}$ and $\widehat{\lambda}_j^{(\gamma')}$ are expressed as Eqs.(47b), (47c) or (47d), $A_{i,j} = 0$. When $\widehat{\lambda}_i^{(\gamma')}$ and $\widehat{\lambda}_j^{(\gamma')}$ are expressed as Eq.(47a),

$$\begin{aligned} A_{i,j} &= \frac{(s_i^2 - \sigma^2)^2 (s_j^2 - \sigma^2)^2}{d_i d_j s_i^2 s_j^2} \int_0^\infty \frac{1}{(1+\gamma')^2} d\gamma' \\ &= \frac{(s_i^2 - \sigma^2)^2 (s_j^2 - \sigma^2)^2}{d_i d_j s_i^2 s_j^2}. \end{aligned} \tag{91}$$

Similarly, when $\widehat{\lambda}_i^{(\gamma')}$ is expressed as Eqs.(47b), (47c) or (47d), $B_i = 0$. When $\widehat{\lambda}_i^{(\gamma')}$ are expressed as Eq.(47a),

$$\begin{aligned} B_i &= \frac{2\sigma^2 (s_i^2 - \sigma^2)^2 (2s_i^2 - \sigma^2)}{d_i^2 s_i^4} \int_0^\infty \frac{1}{(1+\gamma')^2} d\gamma' \\ &\quad - \frac{\sigma^2 (s_i^2 - \sigma^2)^3 (2s_i^2 - \sigma^2)}{d_i^2 s_i^6} \int_0^\infty \frac{1}{(1+\gamma')^3} d\gamma' \\ &= \frac{\sigma^2 (3s_i^2 + \sigma^2)(s_i^2 - \sigma^2)^2 (2s_i^2 - \sigma^2)}{2 d_i^2 s_i^6}. \end{aligned} \tag{92}$$

Then we can express A and B as

$$A = \sum_{i=1}^n \sum_{j=1}^n A_{i,j}, \tag{93}$$

$$B = \sum_{i=1}^n B_i. \tag{94}$$

Below, we give a proof depending on A and B (see Table 3).

(A) If $A > B$ and $B < 0$, Eq.(88) yields $\widehat{J}_E'(\gamma) > 0$ for any $\gamma \in [0, \infty)$. This implies that $\widehat{J}_E(\gamma)$ is monotone increasing and thus $\widehat{\gamma} = 0$.

(B) If $A > B \geq 0$, Eq.(88) implies that $\widehat{J}_E'(\gamma) = 0$, where

$$\widetilde{\gamma} = \frac{B}{A - B} \ (\geq 0). \tag{95}$$

Since

$$\widehat{J}_E(\gamma) - \widehat{J}_E(\tilde{\gamma}) = \frac{4A(\gamma - \tilde{\gamma})^2}{(1+\gamma)^2 (1+\tilde{\gamma})^2} \geq 0, \tag{96}$$

Table 3: Cases in proof of Theorem2.

Conditions		Results
$A > B$	$B < 0$	(A) $\widehat{\gamma} = 0$
	$B \geq 0$	(B) $\widehat{\gamma} = \tilde{\gamma}$
$A < B$		(C) $\widehat{\gamma} = \infty$
$A = B$	$B \neq 0$	(D) $\widehat{\gamma} = \infty$
	$B = 0$	(E) $\widehat{\gamma} \in [0, \infty)$

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

(C) If $A < B$, we have $B > 0$ since $A \geq 0$. Then Eq.(88) yields $\widehat{J}'_E(\gamma) < 0$ for any $\gamma \in [0, \infty)$. This implies that $\widehat{J}_E(\gamma)$ is monotone decreasing and thus $\widehat{\gamma} = \infty$.

(D) If $A = B \neq 0$, we have $B > 0$ since $A > 0$. Then Eq.(88) yields $\widehat{J}'_E(\gamma) < 0$ for any $\gamma \in [0, \infty)$. This implies that $\widehat{J}_E(\gamma)$ is monotone decreasing and thus $\widehat{\gamma} = \infty$.

(E) If $A = B = 0$, Eq.(88) yields $\widehat{J}'_E(\gamma) = 0$. This implies that $\widehat{J}(\gamma)$ is constant so $\widehat{\gamma}$ is an arbitrary value in $[0, \infty)$.

By summarizing the above results (see Table 3), we have Eq.(56). ■

References

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

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

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

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

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

[6] 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.

[7] 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.

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

[9] 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.

[10] M. Sugiyama and K.R. Müller, "The subspace information criterion for infinite dimensional hypothesis spaces," Journal of Machine Learning Research, vol.3, pp.323–359, Nov. 2002.

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

[12] M. Sugiyama and K. Sakurai, "Analytic optimization of shrinkage parameters based on regularized subspace information criterion," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol.E89-A, no.8, pp.2216–2225, 2006.