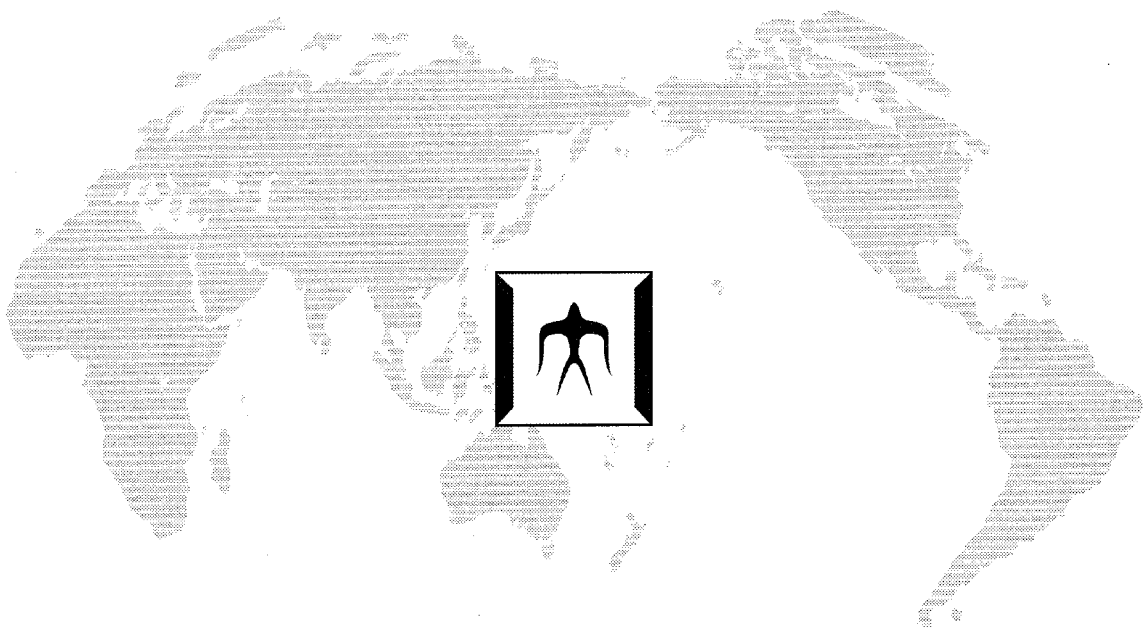
論文 / 著書情報
Article / Book Information

題目(和文)	
Title(English)	A Theory of Model Selection and Active Learning for Supervised Learning
著者(和文)	杉山将
Author(English)	Masashi Sugiyama
出典(和文)	学位:工学博士, 学位授与機関:東京工業大学, 報告番号:甲第4824号, 授与年月日:2001年3月26日, 学位の種別:課程博士, 審査員:小川英光
Citation(English)	Degree:Doctor of Engineering, Conferring organization: Tokyo Institute of Technology, Report number:甲第4824号, Conferred date:2001/3/26, Degree Type:Course doctor, Examiner:
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

A Theory of Model Selection and Active Learning for Supervised Learning



Department of Computer Science,
Graduate School of Information Science and Engineering,
Tokyo Institute of Technology, Japan.

January 2001

Masashi Sugiyama

To my parents

Contents

List of tables

List of figures

Nomenclature

$\langle \cdot, \cdot \rangle$	inner product	$\overline{\cdot}$	complex conjugate
$\| \cdot \|$	norm	$\| \cdot \|$	absolute value
$\cdot \otimes \bar{\cdot}$	Neumann-Schatten product	$[\cdot]_i$	i-th element of a vector
$^\top$	transpose of a vector	$[\cdot]_{i,j}$	(i,j)-th element of a matrix
*	adjoint of an operator	$\mathcal{R}(\cdot)$	range of an operator
†	Moore-Penrose generalized inverse	$\mathcal{N}(\cdot)$	null space of an operator
tr	trace		

$f(\boldsymbol{x})$	learning target function	J_G	generalization error of $\hat{f}$
L	dimension of input vector $\boldsymbol{x}$	X	$\hat{f} = X\boldsymbol{y}$
$\mathcal{D}$	domain of $f(\boldsymbol{x})$	A	$Af = \boldsymbol{z}$
$\boldsymbol{x}_m$	m-th sample point	P	orthogonal projection operator
$\xi_m^{(l)}$	l-th element of $\boldsymbol{x}_m$	Q	noise covariance matrix
ϵ_m	m-th additive noise	σ^2	noise variance
y_m	m-th sample value	B	design matrix
$(\boldsymbol{x}_m, y_m)$	m-th training example	θ	model
M	number of training examples	$\mathcal{M}$	set of models
$\mathcal{X}$	set of training examples	S	subspace of H
$\boldsymbol{z}$	$(f(\boldsymbol{x}_1), f(\boldsymbol{x}_2), \ldots, f(\boldsymbol{x}_M))^\top$	T	regularization operator
$\boldsymbol{\epsilon}$	$(\epsilon_1, \epsilon_2, \ldots, \epsilon_M)^\top$	α	regularization parameter
$\boldsymbol{y}$	$(y_1, y_2, \ldots, y_M)^\top$	μ	dimension of H
$\hat{f}(\boldsymbol{x})$	learning result function	$\varphi_p(\boldsymbol{x})$	p-th basis functions in H
H	reproducing kernel Hilbert space to which $f(\boldsymbol{x})$ belongs	$\mathbf{C}^M$	M-dimensional unitary space
		$\mathrm{E}_{\boldsymbol{\epsilon}}$	expectation over noise $\boldsymbol{\epsilon}$
$K(\boldsymbol{x}, \boldsymbol{x}')$	reproducing kernel of H		

Chapter 1

Introduction

This dissertation is devoted to developing a theory of supervised learning for clarifying the *essence* of learning. In this chapter, we state the motivation and objective of our work.

1.1 Three issues in learning

Learning is obtaining an underlying rule by using information sampled from the environment. If the underlying rule is identified, then it is possible to deal with unknown situations. This ability is called the *generalization capability.* Acquiring a higher level of the generalization capability means that the learning target has been well recognized. The methodology of physics, to clarify the principles of the world of nature from a limited amount of experimental data, is a typical model of learning. Hence, learning is a very important issue in science.

There are three major issues in learning. First, clarifying the mechanism of learning in the brain of human beings. This issue has been mainly studied in psychology, biology, and neuroscience. Second, developing learning machines. This has been studied in computer science and neuroengineering. Third, investigating the *essence* of learning, for example, the generalization capability. This has been mainly studied in the field of information science. Although the three issues are listed independently, each contributes to the development of the others (Figure 1.1). The main goal of our work is to clarify the mechanism of acquiring the generalization capability corresponding to the third issue.

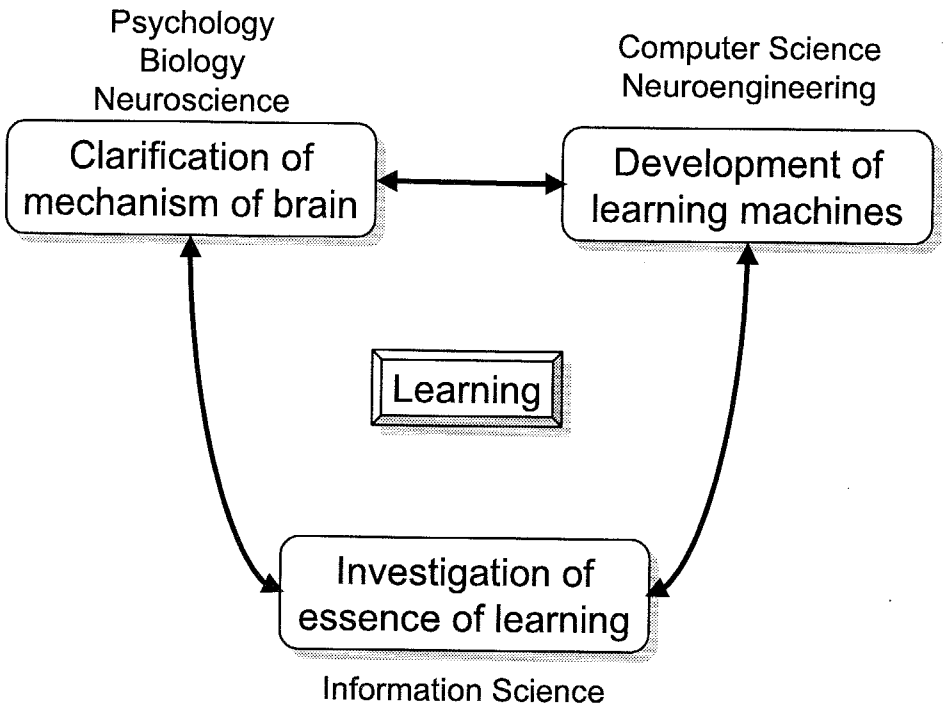


Figure 1.1: Three issues in learning. Although the three issues are listed independently, each contributes to the development of the others.

1.2 Contribution of this dissertation

Our work to clarify the essence of learning contributes to both the clarification of the mechanism of the brain and the development of learning machines. In this section, we briefly review the studies of neuroscience and neuroengineering, and show how our work contributes to these subjects.

1.2.1 Clarification of mechanism of brain

The principles of learning in the brain of human beings is one of the most mysterious issues in science. Activities of the brain can be classified into two different but closely related categories, depending on the type of thinking (Amari [8]). One type of thinking is *logical thinking*, where information is represented by simplified symbols. Human beings achieved the symbolization of information by creating languages. Once information is symbolized, it is consciously and sequentially processed with logical relations of symbols. Although logical thinking is an important activity in the brain, it occupies only a part of its activities. The other type of thinking is *intuitive thinking*, where an enormous amount

of information which is not symbolized but represented by activation patterns on *neurons* is unconsciously processed. In intuitive thinking, information is processed in parallel by the interaction of neurons. Then reasonable solutions are obtained by integrating a huge amount of contradictory information. This process closely relates to the mechanism of memorization in the brain. The interaction of neurons is modified through learning.

Each type of thinking has advantages and disadvantages (Amari [8]). Logical thinking enables us to reason deeply and accurately. However, its ability is poor when it comes to induction and summarization. Moreover, it requires a lot of time to derive a solution and it is easily affected by a lack of or the contradictory information. In contrast, intuitive thinking gives a good, quick response for *ill-posed* problems. The drawback is its inaccuracy. Human beings show their splendid information processing ability by combining these two different types of thinking.

The ultimate purpose of neuroscience is to clarify the principles of the brain, particularly the distributed representation of information and information processing (Kawato [59], Amari [8]). The usual approach is to measure its activities in detail. Our brain has been developed through evolution and mutation. Creatures with successful brain architecture evolved as those less successful were selected against. Owing to the process of the development, our brain became very complicated. As a result, it is extremely difficult to clarify the principles of the brain by measuring its activities. An alternative approach to the clarification of the brain is to assume a model of the brain, i.e., an artificial neural network, and to mathematically study the potentialities and limitations of the model. If the limitations are known, we can improve the architecture of the model, the representation of information, and the learning algorithm. This will lead us to a better understanding of the principles of the brain.

This theoretical approach is generally called *computational neuroscience*. Kawato [59] defined computational neuroscience as follows:

> Computational neuroscience is an approach which aims at clarifying the mechanism of the brain so deeply and essentially as to be able to devise computer programs or machines which realize the function of the brain as the brain does.

In the development of such learning machines, our work in this dissertation will play an essential role. Note that the goal of computational neuroscience is not to make computers but to understand the mechanism of the brain.

Table 1.1: Marr's three levels for understanding the brain (Marr [75]).

Computational theory
What is the purpose of learning?
Why is the purpose suitable?
Representation and algorithm
What representation of input and output is appropriate?
What algorithm achieves the purpose of learning?
Hardware implementation
How are the representation and algorithm realized?

In order to clarify the principles of the brain, Marr [75] insisted on the need for studies from three different levels, *computational theory*, *representation and algorithm*, and *hardware implementation* (Table 1.1). The first level is an abstract computational theory, where the purpose of learning and its suitability are discussed. In the second level, the representation of input and output, and algorithm which maps the input to output are determined. Namely, how to achieve the purpose of learning is inquired. Finally, in the third level, the realization of the representation in hardware and algorithm are studied. Our work in this dissertation will particularly contribute to the second level of Marr's.

1.2.2 Development of learning machines

The basic principles of computers of *von Neumann type* is based on logical thinking. Logical thinking is formalized as symbol processing, where the theories of computational complexity and algorithm in *Turing machines* play a crucial role. The technology of computers has developed dramatically over the last several decades. With the ready availability of computers, the symbol processing approach to artificial intelligence has been extensively studied. However, it has not always succeeded in real world problems, which consist of a great deal of information full of contradiction. For example, in visual information processing, the brain of babies can recognize objects and faces faster and more accurately than the state-of-the-art artificial intelligence systems running in the latest computers (Hertz *et al.* [48]). This motivated many researchers in the engineering field to devise *neurocomputers*, which have the following beneficial features (Hertz *et al.* [48]):

- Neurocomputers can adapt themselves to the new environment. They do not require

programs which prescribe the response to the expected stimuli (events).

- Neurocomputers run in parallel. Namely, each neuron (unit) works asynchronously.
- Neurocomputers can process vague, noisy, and contradictory information.
- Neurocomputers are tolerant of errors. Specifically, they are resistant to faults in neurons (units).
- Neurocomputers are small and efficient in electric power.

The basic principles of neurocomputers are supported by theories of learning. Therefore, studies of the essence of learning, which is the main subject of this dissertation, are indispensable for the development of neurocomputers. Note that neurocomputers do not have to be realized in the form of neural networks. If the above requirements are met, their realization is not restricted (Amari [8]).

1.3 Investigation of essence of learning

We have seen that studies of the essence of learning, i.e., the mechanism of acquiring the generalization capability, are beneficial to both the clarification of the mechanism of the brain and the development of neurocomputers. In this section, we show our focus and explain the problem of supervised learning in detail.

1.3.1 Focus of this dissertation

The investigation of the generalization capability is mainly studied in the following aspects:

- Statistical learning theory,
- Computational learning theory,
- Bayesian learning theory.

In the statistical learning community, the generalization capability is evaluated in the average sense, while the worst generalization capability is analyzed in the computational learning community. The Bayesian learning theory analyzes the posterior probability,

which is different from the generalization capability. In this dissertation, we take the first approach, i.e., we will evaluate the expected generalization capability.

Depending on the type of information made available, learning is classified into *supervised learning* and *unsupervised learning*. In supervised learning, input data and corresponding output data are available. Supervised learning is also referred to as *learning from examples* (Hertz *et al.* [48], Cohn [24]). The purpose of supervised learning is to estimate an underlying rule from training data. On the other hand, only input data and its stochastic properties are available in unsupervised learning. Central topics in unsupervised learning are, for example, associative memory (Kohonen [63]), optimization (Hopfield & Tank [49][50]), density estimation (e.g. Vapnik [140][141]), and principal component analysis (e.g. Jolliffe [57]). In this dissertation, we focus on the former supervised learning.

1.3.2 Supervised learning

Now we explain the problem of supervised learning in detail. Let us consider an underlying input-output system $f(\boldsymbol{x})$. If an input point $\boldsymbol{x}_m$ is given to the system, then an output value $y_m = f(\boldsymbol{x}_m) + \epsilon_m$ is emitted from the system, where ϵ_m is a random additive noise. The purpose of supervised learning is to estimate the underlying function $f(\boldsymbol{x})$. If an approximation $\hat{f}(\boldsymbol{x})$ of the learning target function $f(\boldsymbol{x})$ is successfully acquired, then an estimate $\hat{v}_0$ of an output value v_0 corresponding to a future input value $\boldsymbol{u}_0$ can be obtained as (see Figure 1.2)

$$\hat{v}_0 = \hat{f}(\boldsymbol{u}_0). \tag{1.1}$$

The ability to estimate future output values is called the generalization capability. The generalization capability of a learning result function $\hat{f}(\cdot)$ is evaluated by the *generalization error*, which is typically defined as

$$\mathrm{E}_{\boldsymbol{\epsilon}} \int \left| \hat{f}(\boldsymbol{u}) - f(\boldsymbol{u}) \right|^2 w(\boldsymbol{u}) d\boldsymbol{u}, \tag{1.2}$$

where $\mathrm{E}_{\boldsymbol{\epsilon}}$ denotes the expectation over the noise, and $w(\cdot)$ is a certain weight function, e.g. the probability density function of future sample points $\boldsymbol{u}$.

So far, various methods have been proposed for supervised learning. Most of the methods rely on the availability of a large number of training examples (e.g. Mallows [72][73], Akaike [1], Takeuchi [133], Schwarz [115], Rissanen [99][100][101], Craven & Wahba [28],

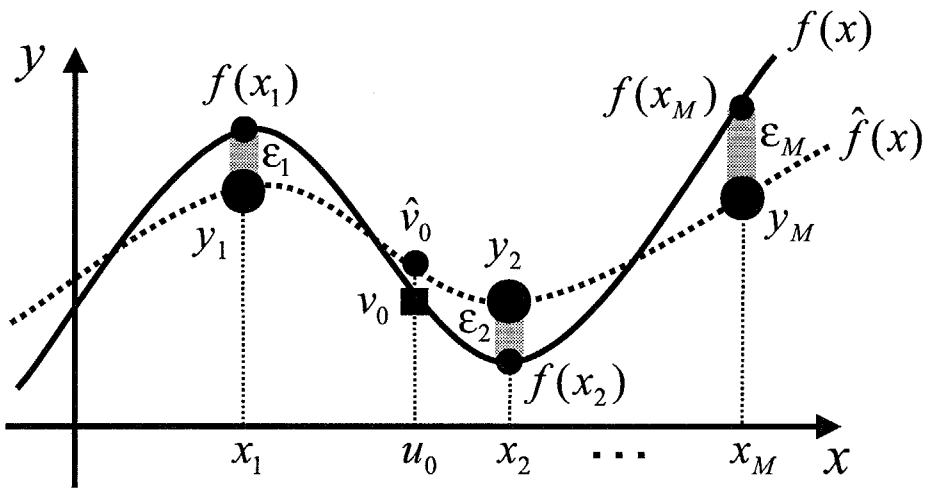


Figure 1.2: Supervised learning. The solid curve denotes the learning target function $f(\boldsymbol{x})$ and the dashed curve denotes a learning result function $\hat{f}(\boldsymbol{x})$.

Rumelhart *et al.* [102], Murata *et al.* [82], Konishi & Kitagawa [64], Ishiguro *et al.* [55], Amari [9]). However, this assumption does not always hold in practice.

One of the main concerns of this dissertation is developing a learning theory that is valid for small sample cases. Various studies in this line can also be found in many articles (e.g. Sugiura [127], Ogawa [89][90], Hurvich and Tsai [52][53][54], Noda *et al.* [85], Fujikoshi and Satoh [36], Satoh *et al.* [107], Vijayakumar and Ogawa [142][143], Vijayakumar *et al.* [144], Hurvich *et al.* [51], Simonoff [121], McQuarrie and Tsai [76], Vapnik [140][141], Cherkassky *et al.* [23], Sugiyama and Ogawa [129][130]).

1.3.3 Three key factors for optimal generalization

In the process of supervised learning, we can control the following three factors for optimal generalization (Figure 1.3).

- A model, which typically indicates a set of functions from which the learning result function $f(\cdot)$ is searched. Figure 1.4 displays learning result functions obtained with three different models. If the model is too simple, then the learning result function is *under-fitted*. If the model is too complex, then the learning result function is *over-fitted*. In general, both over- and under-fitted learning result functions have lower levels of the generalization capability. If the model complexity is chosen appropriately, then a good learning result function is obtained.

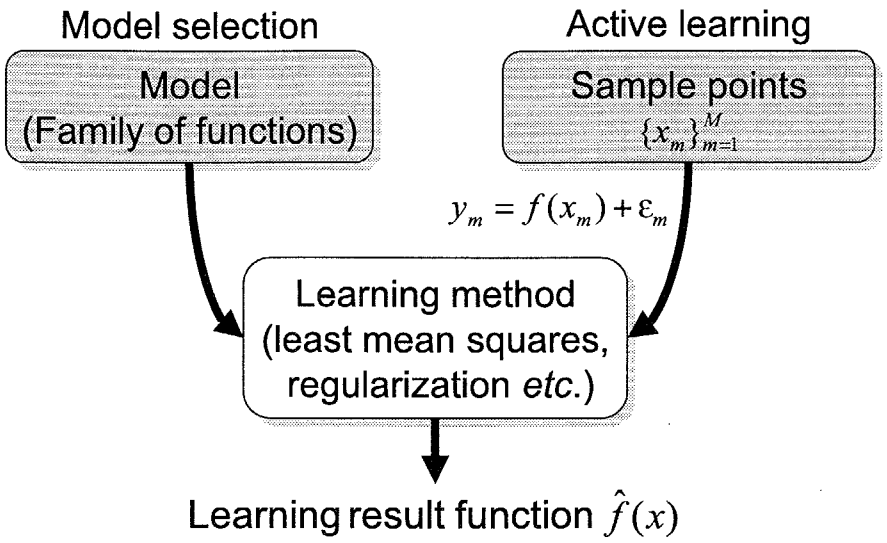
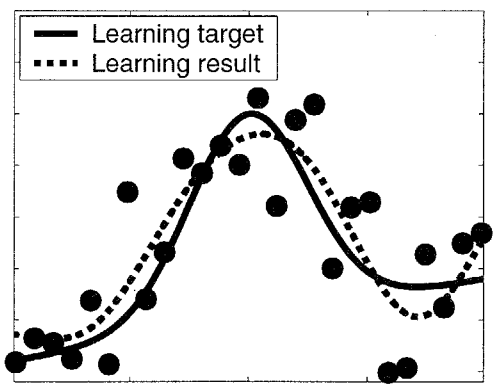
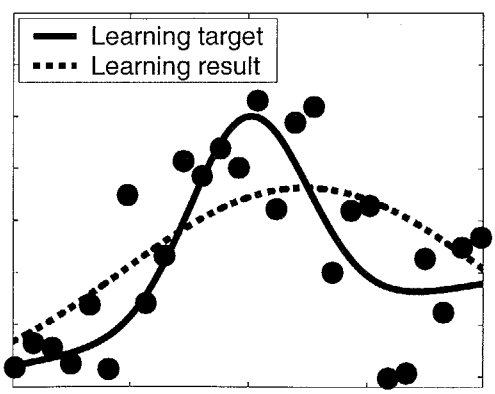


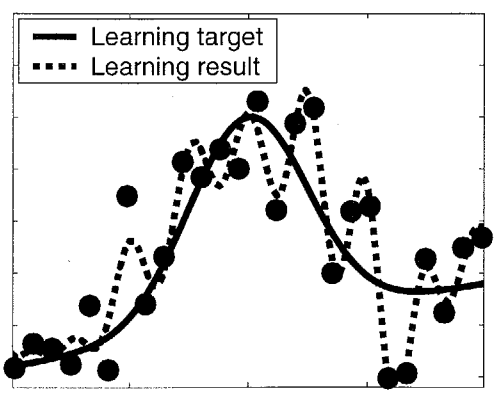
Figure 1.3: Three factors for optimal generalization.



Appropriate model



Simple model



Complex model

Figure 1.4: Learning result functions from different models.

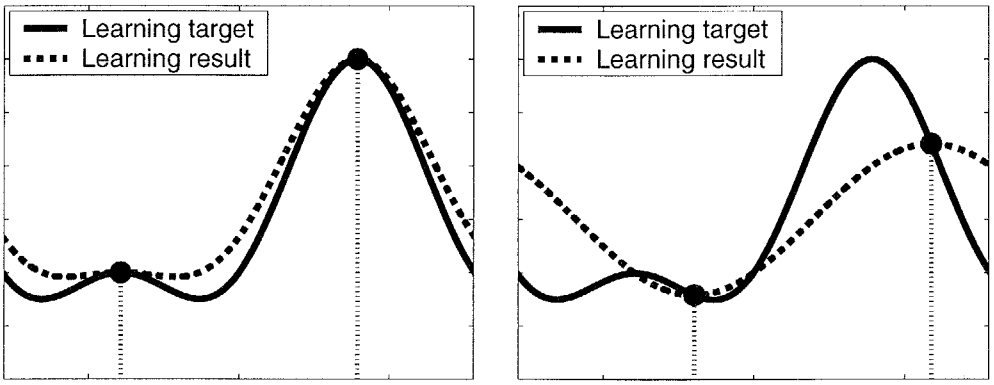


Figure 1.5: Learning result functions from different sets of sample points.

- Sample points $\{\boldsymbol{x}_m\}_{m=1}^M$, at which sample values $\{y_m\}_{m=1}^M$ are gathered. Figure 1.5 displays learning result functions obtained with two different sets of two sample points. The figure shows that the generalization capability depends heavily on the location of sample points. If the sample points are determined appropriately, then a good learning result function is obtained.

- A learning method, which determines a learning result function $\hat{f}(\cdot)$ from a given model and training examples. Generally, a learning method is prescribed by a learning criterion and the learning result function is defined as the minimizer of the learning criterion.

The above three factors are closely related each other. Indeed, one of the factors can not be generally optimized without fixing the others. In this dissertation, we fix the third factor, a learning criterion, and we discuss the following three problems.

- Optimize models with sample points and a learning method fixed (Chapter 4).

- Optimize sample points with a model and a learning method fixed (Chapter 5).

- Optimize models and sample points at the same time with a learning method fixed (Chapter 6).

The first and second problems are called *model selection* and *active learning*, respectively, and we call the third problem *active learning with model selection*, which is a challenging subject.

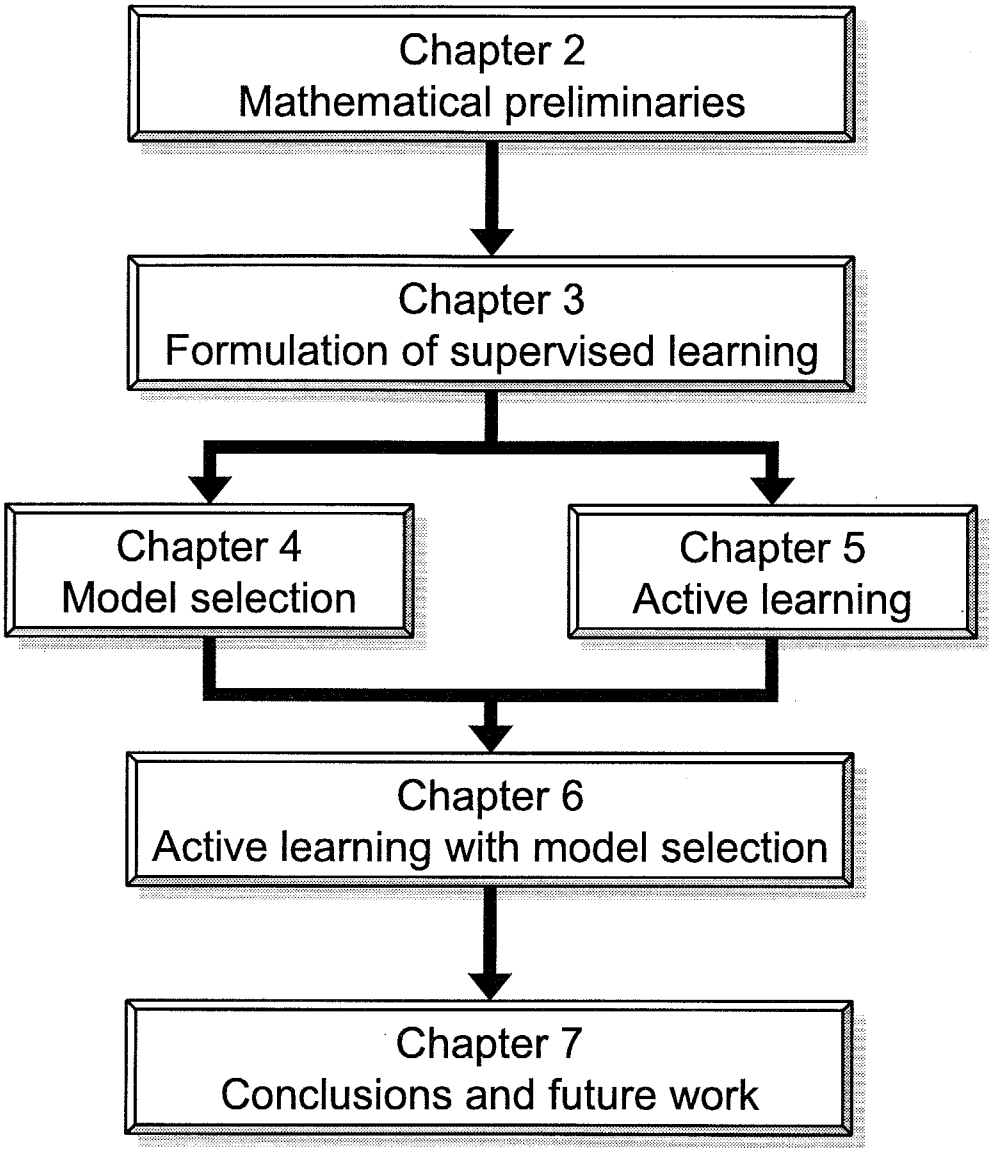


Figure 1.6: Organization of this dissertation.

1.4 Organization of this dissertation

This dissertation consists of seven chapters (Figure 1.6). In this section, we show the organization of this dissertation.

Chapter 2 introduces mathematical concepts, and Chapter 3 mathematically formulates the problem of supervised learning. Some of the major learning methods and models are also reviewed. These two chapters form the basis of this dissertation, so I recommend reading them first.

Within the framework, Chapter 4 and Chapter 5 are devoted to developing theories

of model selection and active learning, respectively. These two chapters are independent so they can be read separately.

In Chapter 4, we discuss the problem of model selection and propose a method of selecting models for the optimal generalization capability. In Section 4.1, the motivations and importance of model selection are explained. In Section 4.2, the model selection problem is mathematically formulated. In Section 4.3, a model selection criterion named the *subspace information criterion (SIC)* is derived. Basic properties and practical expression of SIC are also shown. In Sections 4.4 and 4.5, SIC is applied to least mean squares learning and regularization learning, respectively. In Section 4.6, SIC is compared with a large number of existing model selection techniques. In Section 4.7, an approximation of SIC, which is computationally more efficient than the original SIC, is derived. Its relation to existing methods are also investigated. Section 4.8 is devoted to computer simulations, demonstrating the outstanding performance of SIC in small sample cases. Proofs of all theorems, corollaries, and lemmas given in this chapter are provided in Section 4.9.

In Chapter 5, we discuss the problem of active learning and propose methods of designing sample points for the optimal generalization capability. In Section 5.1, the objectives and necessity of active learning are stated. In Section 5.2, the active learning problem is mathematically formulated. Within this formulation, two kinds of active learning methods are devised. The first is a batch active learning method given in Section 5.3, which specifies all sample points at the same time. The optimality of this method is theoretically proved, and the mechanism of achieving the optimal generalization capability is clarified by using the properties of pseudo orthonormal bases. An efficient calculation method of learning result functions is also provided. The second is an incremental active learning method given in Section 5.4, which specifies sample points one by one. The range of application of the incremental method is wider than that of the batch method. In Section 5.5, the proposed active learning methods are compared with existing active learning methods. In Section 5.6, computer simulations are performed to demonstrate the excellent performance of the proposed methods in small sample cases. Section 5.7 is devoted to reviewing a concept of pseudo orthonormal bases, which plays an important role in Section 5.3. Finally, proofs of all theorems, corollaries, and lemmas given in this chapter are provided in Section 5.8.

So far, the problems of model selection and active learning are independently discussed. In Chapter 6, we consider a more challenging subject: selecting models and

sample points at the same time. We call this subject *active learning with model selection*. The motivations and importance of active learning with model selection are stated in Section 6.1. Here, we point out that the problem of active learning with model selection can not be generally solved by simply combining existing active learning and model selection methods because of the *active learning / model selection dilemma*, i.e., the model should be fixed for active learning and conversely the sample points should be fixed for model selection. In Section 6.2, the problem of active learning with model selection is mathematically formulated. In Section 6.3, a basic strategy for avoiding the dilemma is given. Based on the strategy, a procedure for simultaneously optimizing sample points and models is devised in Section 6.4. An efficient and fast algorithm for the procedure is also provided. In Section 6.5, the effectiveness of the proposed algorithm is experimentally demonstrated through computer simulations. Proofs of all theorems and lemmas given in this chapter are provided in Section 6.6.

Finally, conclusions of this dissertation and future work are stated in Chapter 7.

Chapter 2

Mathematical preliminaries

This chapter briefly reviews the mathematical concepts used in this dissertation.

2.1 Hilbert space notation

Let $\langle \cdot, \cdot \rangle$ be the *inner product* in a Hilbert space H. The inner product has the following properties for elements f and g in H and a scalar a:

$$\langle a_1 f_1 + a_2 f_2, g \rangle = a_1 \langle f_1, g \rangle + a_2 \langle f_2, g \rangle, \tag{2.1}$$

$$\langle f, g \rangle = \overline{\langle g, f \rangle}, \tag{2.2}$$

$$\langle f, f \rangle \geq 0, \quad \text{if } \langle f, f \rangle = 0 \text{ then } f = 0. \tag{2.3}$$

where $\overline{\cdot}$ denotes the complex conjugate of a scalar.

Let $\| \cdot \|$ be the *norm* in the Hilbert space H defined as

$$\|f\| = \sqrt{\langle f, f \rangle}. \tag{2.4}$$

The norm has the following properties:

$$\|f\| \geq 0, \quad \|f\| = 0 \text{ if and only if } f = 0, \tag{2.5}$$

$$\|af\| = |a| \, \|f\|, \tag{2.6}$$

$$\|f + g\| \leq \|f\| + \|g\|, \tag{2.7}$$

$$|\langle f, g \rangle| \leq \|f\| \, \|g\|. \tag{2.8}$$

2.2 Linear operators

Here, we show the notation of linear operators and review some distinctive linear operators. From here on, the term 'operator' is used as 'linear operator'.

2.2.1 Adjoint operators

Let A be an operator from a Hilbert space H_1 to a Hilbert space H_2. The operator A^* is called the *adjoint operator* of A if it satisfies

$$\langle Af, g\rangle = \langle f, A^*g\rangle \text{ for any } f \in H_1, \text{ any } g \in H_2. \tag{2.9}$$

The adjoint operator has the following properties:

$$(A^*)^* = A, \tag{2.10}$$

$$(a_1A_1 + a_2A_2)^* = \overline{a}A_1^* + \overline{a_2}A_2^* \text{ for any scalars } a_1, a_2, \tag{2.11}$$

$$(A_1A_2)^* = A_2^*A_1^*, \tag{2.12}$$

$$(A^{-1})^* = (A^*)^{-1}. \tag{2.13}$$

A is called the *self-adjoint operator* if it satisfies

$$A^* = A. \tag{2.14}$$

2.2.2 Range and null space

For an operator A from a Hilbert space H_1 to a Hilbert space H_2, let us denote the *range* and *null space* of A by $\mathcal{R}(A)$ and $\mathcal{N}(A)$, respectively:

$$\mathcal{R}(A) = \{g \mid g = Af \text{ for all } f \in H_1\}, \tag{2.15}$$

$$\mathcal{N}(A) = \{f \mid Af = 0 \text{ for all } f \in H_1\}. \tag{2.16}$$

Let $\mathcal{R}(A)^\perp$ be the *orthogonal complement* of $\mathcal{R}(A)$. Then the following relations hold:

$$\mathcal{R}(A)^\perp = \mathcal{N}(A^*), \tag{2.17}$$

$$\mathcal{R}(A^*)^\perp = \mathcal{N}(A), \tag{2.18}$$

$$\mathcal{N}(A^*A) = \mathcal{N}(A), \tag{2.19}$$

$$\mathcal{R}(A^*A) = \mathcal{R}(A^*). \tag{2.20}$$

2.2.3 Distinctive operators

For an operator A from a Hilbert space H_1 to a Hilbert space H_2, A is called the *unitary operator* if it satisfies

$$A^* = A^\dagger. \tag{2.21}$$

A is called an *isometry* if it satisfies

$$\|Af\| = \|f\| \text{ for any } f \in H_1. \tag{2.22}$$

A is called a *partial isometry* if it satisfies

$$\|Af\| = \begin{cases} \|f\| & \text{for any } f \in \mathcal{N}(A)^{\perp}, \\ 0 & \text{for any } f \in \mathcal{N}(A). \end{cases} \tag{2.23}$$

For an operator A from a Hilbert space H to the same Hilbert space H, A is said to be *non-negative* or *positive semidefinite* if it satisfies

$$\langle Af, f \rangle \geq 0 \text{ for any } f \in H. \tag{2.24}$$

A is said to be *positive* or *positive definite* if

$$\langle Af, f \rangle > 0 \text{ for any } f \neq 0. \tag{2.25}$$

Let P_S be the *orthogonal projection operator* onto a subspace S:

$$P_S f = \begin{cases} f & \text{for any } f \in S, \\ 0 & \text{for any } f \in S^{\perp}. \end{cases} \tag{2.26}$$

The orthogonal projection operator has the following properties:

$$P_S^2 = P_S, \tag{2.27}$$

$$P_S^* = P_S. \tag{2.28}$$

2.2.4 Neumann-Schatten product

For an element g in a Hilbert space H_1 and an element f in a Hilbert space H_2, the *Neumann-Schatten product* $(f \otimes \overline{g})$ is an operator from H_1 to H_2 defined by using any h in H_1 as (Schatten [111])

$$(f \otimes \overline{g})\, h = \langle h, g \rangle f. \tag{2.29}$$

The Neumann-Schatten product has the following properties:

$$a\,(f \otimes \overline{g}) = (af) \otimes \overline{g} = f \otimes \overline{(\overline{a} g)} \text{ for any scalar } a, \tag{2.30}$$

$$(f_1 + f_2) \otimes \overline{g} = f_1 \otimes \overline{g} + f_2 \otimes \overline{g}, \tag{2.31}$$

$$f \otimes \overline{(g_1 + g_2)} = f \otimes \overline{g_1} + f \otimes \overline{g_2}, \tag{2.32}$$

$$A\left(f \otimes \overline{g}\right) = (Af) \otimes \overline{g} \text{ for any operator } A, \tag{2.33}$$

$$\left(f \otimes \overline{g}\right) A = f \otimes \overline{(A^* g)} \text{ for any operator } A, \tag{2.34}$$

$$\left(f \otimes \overline{g_1}\right)\left(g_2 \otimes \overline{h}\right) = \langle g_2, g_1 \rangle \left(f \otimes \overline{h}\right), \tag{2.35}$$

$$\left(f \otimes \overline{g}\right)^* = \left(g \otimes \overline{f}\right), \tag{2.36}$$

$$\|f \otimes \overline{g}\| = \|f\| \, \|g\|. \tag{2.37}$$

2.2.5 Moore-Penrose generalized inverse

An operator X that satisfies the following four conditions is called the *Moore-Penrose generalized inverse* of an operator A (see e.g. Albert [5], Ben-Israel & Greville [14]).

$$AXA = A, \tag{2.38}$$

$$XAX = X, \tag{2.39}$$

$$(AX)^* = AX, \tag{2.40}$$

$$(XA)^* = XA. \tag{2.41}$$

Note that the Moore-Penrose generalized inverse is unique if it exists, and it is denoted by $A^\dagger$. The Moore-Penrose generalized inverse has the following properties:

$$\mathcal{R}(AA^\dagger) = \mathcal{R}(AA^*) = \mathcal{R}(A), \tag{2.42}$$

$$\mathcal{N}(AA^\dagger) = \mathcal{N}(AA^*) = \mathcal{N}(A^*) = \mathcal{N}(A^\dagger), \tag{2.43}$$

$$\mathcal{R}(A^\dagger A) = \mathcal{R}(A^* A) = \mathcal{R}(A^*) = \mathcal{R}(A^\dagger), \tag{2.44}$$

$$\mathcal{N}(A^\dagger A) = \mathcal{N}(A^* A) = \mathcal{N}(A), \tag{2.45}$$

$$(A^\dagger)^\dagger = A, \tag{2.46}$$

$$(A^\dagger)^* = (A^*)^\dagger, \tag{2.47}$$

$$(aA)^\dagger = a^\dagger A^\dagger \text{ for any scalar } a, \tag{2.48}$$

$$A^\dagger = (A^* A)^\dagger A^* = A^*(AA^*)^\dagger, \tag{2.49}$$

$$A^* = A^* A A^\dagger = A^\dagger A A^*, \tag{2.50}$$

$$(A^* A)^\dagger = A^\dagger (A^*)^\dagger, \tag{2.51}$$

$$(UAV)^\dagger = V^* A^\dagger U^* \text{ for any unitary operators } U, V, \tag{2.52}$$

$$P^\dagger = P \text{ for any orthogonal projection operator } P. \tag{2.53}$$

2.2.6 Trace

Let μ be the dimension of a Hilbert space H, and $\{\varphi_p\}_{p=1}^{\mu}$ be an orthonormal basis in H. For an operator A from H to H, the *trace* of A (denoted by '$\mathrm{tr}A$') is defined as

$$\mathrm{tr}A = \sum_{p=1}^{\mu} \langle A\varphi_p, \varphi_p \rangle. \tag{2.54}$$

If μ is finite, then the trace always exists. Otherwise, the trace exists if and only if Eq.(2.54) converges. When the trace exists, it is invariant under all orthonormal bases in H, and it has the following properties:

$$\mathrm{tr}A^* = \overline{\mathrm{tr}A}, \tag{2.55}$$

$$\mathrm{tr}\,(aA) = a\,\mathrm{tr}A \ \text{ for any scalar } a, \tag{2.56}$$

$$\mathrm{tr}\,(A_1 + A_2) = \mathrm{tr}A_1 + \mathrm{tr}A_2, \tag{2.57}$$

$$\mathrm{tr}A_1A_2 = \mathrm{tr}A_2A_1, \tag{2.58}$$

$$\mathrm{tr}\,(f \otimes \overline{g}) = \langle f, g \rangle \ \text{ for any } f, g \in H. \tag{2.59}$$

2.3 Reproducing kernel Hilbert space

Let H be a functional Hilbert space and let $\mathcal{D}$ be the domain of the functions in H. The *reproducing kernel* $K(\boldsymbol{x}, \boldsymbol{x}')$ is a bivariate function defined on $\mathcal{D} \times \mathcal{D}$ that satisfies the following conditions.

- For any fixed $\boldsymbol{x}'$ in $\mathcal{D}$, $K(\boldsymbol{x}, \boldsymbol{x}')$ belongs to H as a function of $\boldsymbol{x}$.
- It holds that

$$\langle f(\cdot), K(\cdot, \boldsymbol{x}') \rangle = f(\boldsymbol{x}') \ \text{ for any } f \in H, \text{any } \boldsymbol{x}' \in \mathcal{D}. \tag{2.60}$$

The reproducing kernel has the following properties:

$$K(\boldsymbol{x}, \boldsymbol{x}') = \overline{K(\boldsymbol{x}', \boldsymbol{x})} \ \text{ for any } \boldsymbol{x}, \boldsymbol{x}' \in \mathcal{D}, \tag{2.61}$$

$$K(\boldsymbol{x}, \boldsymbol{x}) \geq 0 \ \text{ for any } \boldsymbol{x} \in \mathcal{D}. \tag{2.62}$$

A Hilbert space that has the reproducing kernel is called the *reproducing kernel Hilbert space.*

A function is treated as a point in general functional Hilbert spaces. Therefore, the value of a function at a point can not be discussed. However, in the reproducing kernel Hilbert space, it is possible to treat the value of a function at a point.

Let μ be the dimension of H. Then the reproducing kernel of H can be expressed by using an orthonormal basis $\{\varphi_p(\boldsymbol{x})\}_{p=1}^{\mu}$ in H as

$$K(\boldsymbol{x}, \boldsymbol{x}') = \sum_{p=1}^{\mu} \varphi_p(\boldsymbol{x})\overline{\varphi_p(\boldsymbol{x}')}. \tag{2.63}$$

If μ is finite, then the reproducing kernel always exists. Otherwise, the reproducing kernel exists if and only if Eq.(2.63) converges. When the reproducing kernel exists, it is determined uniquely and it is invariant under all orthonormal bases in H. Various properties of the reproducing kernel have been investigated in detail (see e.g. Aronszajn [11], Bergman [16], Saitoh [104][105], Wahba [145], Girosi [42]).

Chapter 3

Formulation of supervised learning

3.1 Functional analytic framework for supervised learning

Let us consider the supervised learning problem of obtaining an approximation to a target function from a set of *training examples*. Let the learning target function $f(\boldsymbol{x})$ be a complex function of L variables defined on a subset $\mathcal{D}$ of the L-dimensional Euclidean space $\mathbf{R}^L$. The training examples are made up of *sample points* $\boldsymbol{x}_m$ in the domain $\mathcal{D}$ and corresponding *sample values* y_m in $\mathbf{C}$:

$$\{(\boldsymbol{x}_m, y_m) \mid y_m = f(\boldsymbol{x}_m) + \epsilon_m\}_{m=1}^{M}, \tag{3.1}$$

where y_m is degraded by additive noise ϵ_m.

The goal of supervised learning is to obtain the optimal approximation $\hat{f}(\boldsymbol{x})$ that minimizes a certain generalization measure J_G. In this dissertation, we assume that the learning target function $f(\boldsymbol{x})$ and a learning result function $\hat{f}(\boldsymbol{x})$ belong to a specified reproducing kernel Hilbert space H (see Section 2.3), and the generalization error of $\hat{f}(\boldsymbol{x})$ is measured by

$$J_G = \mathrm{E}_{\boldsymbol{\epsilon}} \|\hat{f} - f\|^2, \tag{3.2}$$

where $\mathrm{E}_{\boldsymbol{\epsilon}}$ denotes the expectation over the noise and $\|\cdot\|$ denotes the norm in H. The norm is typically defined as

$$\|\hat{f} - f\|^2 = \int \left|\hat{f}(\boldsymbol{u}) - f(\boldsymbol{u})\right|^2 w(\boldsymbol{u}) d\boldsymbol{u}, \tag{3.3}$$

where the integral with respect to $\boldsymbol{u}$ means the expectation over future sample points $\boldsymbol{u}$ and $w(\boldsymbol{u})$ is some weight function, e.g., the probability density function of $\boldsymbol{u}$.

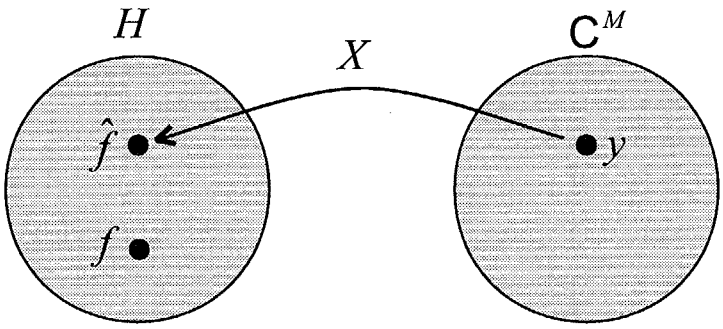


Figure 3.1: Learning framework.

3.2 Learning methods

The learning result function $\hat{f}(\boldsymbol{x})$ is generally obtained by a certain learning method. A learning method is prescribed by a learning criterion and the learning result function is defined as the minimizer of the learning criterion. In this dissertation, we focus on the case when the learning result function $\hat{f}(\boldsymbol{x})$ is given as

$$\hat{f} = X\boldsymbol{y}. \tag{3.4}$$

Here, X is a linear operator from $\mathbf{C}^M$ to H (Figure 3.1) and $\boldsymbol{y}$ is the M-dimensional vector with the m-th element being y_m:

$$\boldsymbol{y} = (y_1, y_2, \ldots, y_M)^\top, \tag{3.5}$$

where $\top$ denotes the transpose of a vector. When the operator X is linear, determining X means determining the coefficients $\{w_p\}_p$ of a linear regression model:

$$\hat{f}(\boldsymbol{x}) = \sum_p w_p \varphi_p(\boldsymbol{x}), \tag{3.6}$$

where $\{\varphi_p(\boldsymbol{x})\}_p$ are prefixed basis functions that span H. In this section, we review *least mean squares learning* and *regularization learning*.

3.2.1 Least mean squares learning

Least mean squares (LMS) learning is one of the most widely used learning criteria defined as follows.

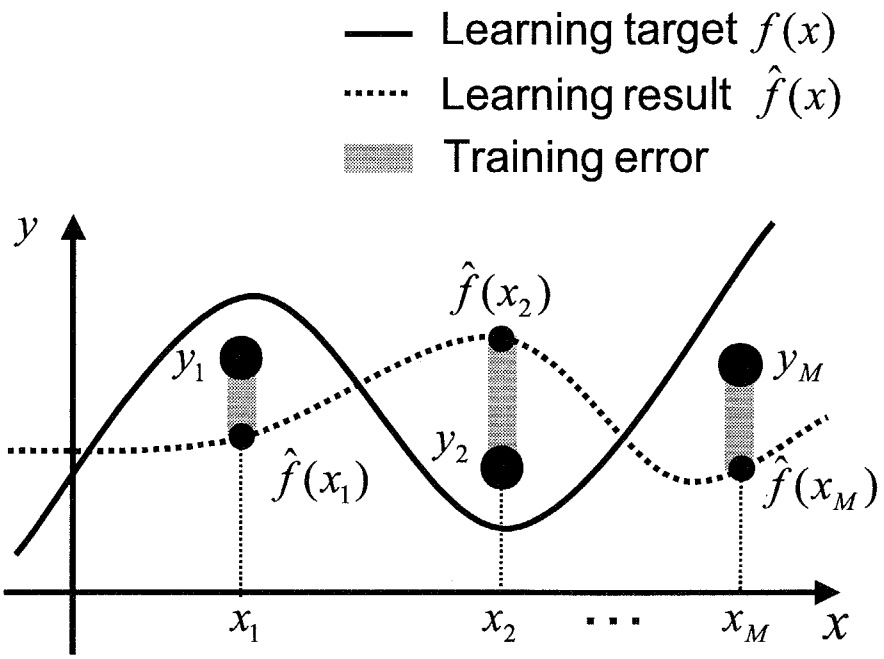


Figure 3.2: Training error.

Definition 3.1 (Least mean squares learning) *Let S be a reproducing kernel Hilbert space included in H. The function $\hat{f}_S(\boldsymbol{x})$ is called the LMS learning function for S if $\hat{f}_S(\boldsymbol{x})$ is the minimizer of the training error J_{TE} in the subspace S:*

$$\hat{f}_S = \operatorname*{argmin}_{\hat{f} \in S} J_{TE}[\hat{f}], \tag{3.7}$$

where

$$J_{TE}[\hat{f}] = \frac{1}{M} \sum_{m=1}^{M} \left| \hat{f}(\boldsymbol{x}_m) - y_m \right|^2. \tag{3.8}$$

Training error is the error at sample points contained in the training set (Figure 3.2). Note that maximum likelihood estimation with Gaussian noise agrees with LMS learning. Let us denote the reproducing kernel of S by $K_S(\boldsymbol{x}, \boldsymbol{x}')$, and let A_S be an operator from H to the M-dimensional unitary space $\mathbf{C}^M$ defined as

$$A_S = \sum_{m=1}^{M} \left(\boldsymbol{e}_m \otimes \overline{K_S(\cdot, \boldsymbol{x}_m)} \right), \tag{3.9}$$

where $(\cdot \otimes \bar{\cdot})$ denotes the Neumann-Schatten product (see Section 2.2.4) and $\boldsymbol{e}_m$ is the m-th vector of the so-called standard basis in $\mathbf{C}^M$. Then the LMS learning function $\hat{f}_S(\boldsymbol{x})$

is given as (see Ogawa [90])

$$\hat{f}_S = X_S \boldsymbol{y}, \tag{3.10}$$

where

$$X_S = A_S^{\dagger}. \tag{3.11}$$

Here, † denotes the Moore-Penrose generalized inverse (see Section 2.2.5).

In the neural network community, gradient-based learning algorithms such as stochastic gradient descent (Amari [7]), back-propagation (Rumelhart *et al.* [102][103]), and natural gradient descent (Amari [9]) are often used. These methods are aimed at finding a LMS learning result function for non-linear regression models such as multi-layer perceptrons, while Eq.(3.10) is valid only for linear regression models given by Eq.(3.6).

3.2.2 Regularization learning

The following regularization learning is also widely used for obtaining a learning function from training examples.

Definition 3.2 (Regularization learning) *Let T be a linear, closed range operator from H to H' and α be a positive constant. The function $\hat{f}_{T,\alpha}(\boldsymbol{x})$ is called the regularization learning function for (T, α) if $\hat{f}_{T,\alpha}(\boldsymbol{x})$ is the minimizer of J_R in H:*

$$\hat{f}_{T,\alpha} = \operatorname*{argmin}_{\hat{f} \in H} J_R[\hat{f}], \tag{3.12}$$

where

$$J_R[\hat{f}] = \sum_{m=1}^{M} \left| \hat{f}(\boldsymbol{x}_m) - y_m \right|^2 + \alpha \|T\hat{f}\|^2, \tag{3.13}$$

and $\|\cdot\|$ is the norm in H'.

The operator T is called the *regularization operator* and the constant α is called the *regularization parameter*. $\|T\hat{f}\|^2$ is called the *regularization term*. Intuitively, the regularization term works so that the learning result function becomes *smooth*, and as a result, over-fitting is avoided. The following regularization operator is often used:

- $T = I_H$, where I_H is the identity operator on H.
- $T = D^n$, where D^n is the n-th derivative operator in H.

- $T = W^{-1}$, where W is defined as

$$W = \sum_{p=1}^{\mu} (\varphi_p \otimes \overline{e_p}) . \tag{3.14}$$

Here, $\{\varphi_p(\boldsymbol{x})\}_{p=1}^{\mu}$ are prefixed μ linearly independent (generally non-orthogonal) functions that span H, e_p is the p-th vector of the so-called standard basis in $\mathbf{C}^{\mu}$, and $(\cdot \otimes \bar{\cdot})$ denotes the Neumann-Schatten product. This type of regularization is called the *weight decay* (Hertz *et al.* [48], Bishop [17]) since the regularization term is expressed as

$$\|T\hat{f}\|^2 = \sum_{p=1}^{\mu} |w_p|^2, \tag{3.15}$$

where $\{w_p\}_{p=1}^{\mu}$ are the coefficients of a linear regression model (see Eq.(3.6)).

Note that regularization learning with $T = W^{-1}$ is equivalent to that with $T = I_H$ if the basis functions $\{\varphi_p(\boldsymbol{x})\}_{p=1}^{\mu}$ are orthonormal:

$$\|W^{-1}\hat{f}\|^2 = \|\hat{f}\|^2. \tag{3.16}$$

The above definition of regularization learning is precisely called *regularization learning with quadratic regularizers* since the regularization term is a quadratic function of $\hat{f}$ (see e.g. Williams [147], Tsuda *et al.* [137] for non-quadratic regularizers).

Let A be an operator from H to the M-dimensional unitary space $\mathbf{C}^M$ defined as

$$A = \sum_{m=1}^{M} \left(\boldsymbol{e}_m \otimes \overline{K(\cdot, \boldsymbol{x}_m)} \right), \tag{3.17}$$

where $\boldsymbol{e}_m$ is the m-th vector of the so-called standard basis in $\mathbf{C}^M$ and $K(\boldsymbol{x}, \boldsymbol{x}')$ is the reproducing kernel of H. Then the regularization learning function $\hat{f}_{T,\alpha}(\boldsymbol{x})$ for the regularization operator T and regularization parameter α is given as (see Nakashima & Ogawa [83])

$$\hat{f}_{T,\alpha} = X_{T,\alpha}\boldsymbol{y}, \tag{3.18}$$

where

$$X_{T,\alpha} = (A^*A + \alpha T^*T)^{\dagger}A^*, \tag{3.19}$$

and A^* is the adjoint operator of A.

Note that we can consider an extended definition of regularization learning that $\hat{f}_{T,\alpha}(\boldsymbol{x})$ is searched in the subspace S of the functional Hilbert space H. However, for the sake of simplicity, we only consider the case when $S = H$ in this dissertation.

3.3 Examples of reproducing kernel Hilbert spaces

In our framework, the learning target function $f(\boldsymbol{x})$ is assumed to belong to a specified reproducing kernel Hilbert space H. As shown in Section 2.3, the reproducing kernel $K(\boldsymbol{x}, \boldsymbol{x}')$ is generally expressed by using an orthonormal basis $\{\varphi_p(\boldsymbol{x})\}_{p=1}^{\mu}$ as

$$K(\boldsymbol{x}, \boldsymbol{x}') = \sum_{p=1}^{\mu} \varphi_p(\boldsymbol{x})\overline{\varphi_p(\boldsymbol{x}')}, \tag{3.20}$$

where $\overline{\cdot}$ denotes the complex conjugate of a scalar. In this section, we review the *trigonometric polynomial space* and *polynomial space* as examples of convenient reproducing kernel Hilbert spaces where the reproducing kernel is expressed in the closed form.

3.3.1 Trigonometric polynomial space

Let us start from the case when the dimension L of the input vector $\boldsymbol{x}$ is 1. The one-dimensional trigonometric polynomial space is defined as follows.

Definition 3.3 (One-dimensional trigonometric polynomial space) *A function space is called a trigonometric polynomial space of order N if it is spanned by the functions*

$$\left\{ \exp(inx) \;\middle|\; n = -N, -N+1, \ldots, N \right\} \tag{3.21}$$

defined on $\mathcal{D} = [-\pi, \pi]$, and the inner product is defined by

$$\langle f, g \rangle = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x)\overline{g(x)} dx. \tag{3.22}$$

The dimension of a trigonometric polynomial space of order N is $2N + 1$, and the reproducing kernel of this space is expressed by

$$K(x, x') = \begin{cases} \sin\dfrac{(2N+1)(x-x')}{2} \Big/ \sin\dfrac{x-x'}{2} & \text{if } x \neq x', \\ 2N+1 & \text{if } x = x'. \end{cases} \tag{3.23}$$

Now we consider multi-dimensional cases. Let us denote the L-dimensional input vector $\boldsymbol{x}$ by

$$\boldsymbol{x} = (\xi^{(1)}, \xi^{(2)}, \ldots, \xi^{(L)})^\top. \tag{3.24}$$

Then the multi-dimensional trigonometric polynomial space is defined as follows.

Definition 3.4 (Trigonometric polynomial space) *For $l = 1, 2, \ldots, L$, let N_l be a non-negative integer and $\mathcal{D}_l = [-\pi, \pi]$. Let*

$$\mathcal{D} = \mathcal{D}_1 \times \mathcal{D}_2 \times \cdots \times \mathcal{D}_L. \tag{3.25}$$

Then a function space is called a trigonometric polynomial space of order $(N_1, N_2, \ldots, N_L)$ if it is spanned by the functions

$$\left\{ \prod_{l=1}^{L} \exp(in_l \xi^{(l)}) \;\middle|\; n_l = -N_l, -N_l + 1, \ldots, N_l \quad \text{for } l = 1, 2, \ldots, L \right\} \tag{3.26}$$

defined on $\mathcal{D}$, and the inner product is defined by

$$\langle f, g \rangle = \frac{1}{(2\pi)^L} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \cdots \int_{-\pi}^{\pi} f(\boldsymbol{x}) \overline{g(\boldsymbol{x})} d\xi^{(1)} d\xi^{(2)} \cdots d\xi^{(L)}. \tag{3.27}$$

The dimension of a trigonometric polynomial space of order $(N_1, N_2, \ldots, N_L)$ is

$$\prod_{l=1}^{L} (2N_l + 1), \tag{3.28}$$

and the reproducing kernel of this space is expressed by

$$K(\boldsymbol{x}, \boldsymbol{x}') = \prod_{l=1}^{L} K_l(\xi^{(l)}, \xi^{(l)\prime}), \tag{3.29}$$

where

$$K_l(\xi^{(l)}, \xi^{(l)\prime}) = \begin{cases} \sin \dfrac{(2N_l + 1)(\xi^{(l)} - \xi^{(l)\prime})}{2} \Big/ \sin \dfrac{\xi^{(l)} - \xi^{(l)\prime}}{2} & \text{if } \xi^{(l)} \neq \xi^{(l)\prime}, \\ 2N_l + 1 & \text{if } \xi^{(l)} = \xi^{(l)\prime}. \end{cases} \tag{3.30}$$

The profile of the reproducing kernel of a trigonometric polynomial space is illustrated in Figure 3.3.

3.3.2 Polynomial space

We again start from the case when the dimension L of the input vector $\boldsymbol{x}$ is 1. The one-dimensional polynomial space is defined as follows.

Definition 3.5 (One-dimensional polynomial space) *A function space is called a polynomial space of order N if it is spanned by the functions*

$$\left\{ x^n \;\middle|\; n = 0, 1, \ldots, N \right\} \tag{3.31}$$

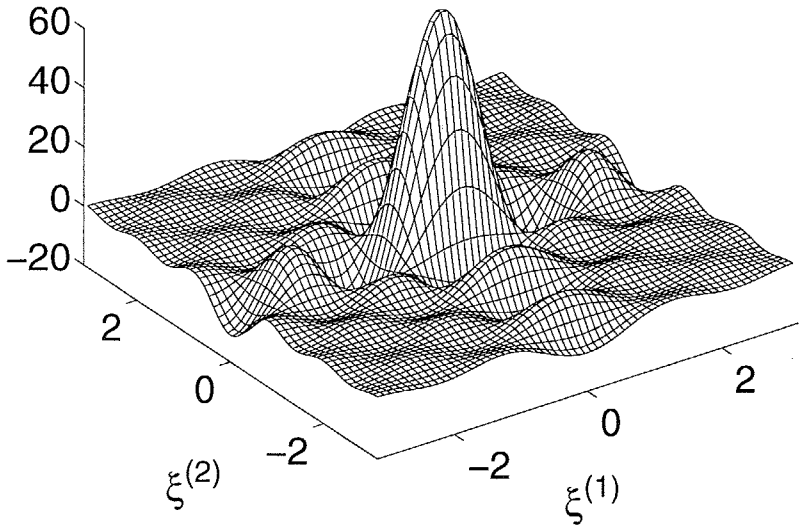


Figure 3.3: Profile of reproducing kernel $K(\boldsymbol{x}, \boldsymbol{x}')$ of a trigonometric polynomial space of order $(3, 5)$ with $\boldsymbol{x}' = (0, 0)^\top$.

defined on $\mathcal{D} = [a, b]$, *and the inner product is defined by*

$$\langle f, g \rangle = \int_a^b f(x)\overline{g(x)}w(x)dx, \tag{3.32}$$

where $w(x)$ *is a weight function such that*

$$w(x) > 0 \quad \text{for } a \leq x \leq b. \tag{3.33}$$

The dimension of a trigonometric polynomial space of order N is $N+1$. Let $\{p_n(x)\}_{n=0}^N$ be orthogonal polynomials expressed by

$$p_n(x) = k_n x^n + k'_n x^{n-1} + \cdots. \tag{3.34}$$

Then it follows from the Christoffel-Darboux formula (see e.g. Szegö [131]) that the reproducing kernel of this space is expressed by

$$K(x, x') = \begin{cases} \dfrac{k_N}{k_{N+1}} \dfrac{p_{N+1}(x)p_N(x') - p_N(x)p_{N+1}(x')}{x - x'} & \text{if } x \neq x', \\ \dfrac{k_N}{k_{N+1}} \left(p'_{N+1}(x)p_N(x') - p'_N(x)p_{N+1}(x')\right) & \text{if } x = x'. \end{cases} \tag{3.35}$$

Now we give the definition of the polynomial space in multi-dimensional cases.

Definition 3.6 (Polynomial space) *For $l = 1, 2, \ldots, L$, let N_l be a non-negative integer and $\mathcal{D}_l = [a_l, b_l]$. Let*

$$\mathcal{D} = \mathcal{D}_1 \times \mathcal{D}_2 \times \cdots \times \mathcal{D}_L. \tag{3.36}$$

Then a function space is called a polynomial space of order $(N_1, N_2, \ldots, N_L)$ if it is spanned by the functions

$$\left\{ \prod_{l=1}^{L} \left(\xi^{(l)}\right)^{n_l} \;\middle|\; n_l = 0, 1, \ldots, N_l \quad \text{for } l = 1, 2, \ldots, L \right\} \tag{3.37}$$

defined on $\mathcal{D}$, and the inner product is defined by

$$\langle f, g \rangle = \int_{a_L}^{b_L} \int_{a_{L-1}}^{b_{L-1}} \cdots \int_{a_1}^{b_1} f(\boldsymbol{x}) \overline{g(\boldsymbol{x})} w(\boldsymbol{x}) d\xi^{(1)} d\xi^{(2)} \cdots d\xi^{(L)}, \tag{3.38}$$

where $w(\boldsymbol{x})$ is a weight function such that

$$w(\boldsymbol{x}) > 0 \quad \text{for } a_l \leq \xi^{(l)} \leq b_l, \;\; l = 1, 2, \ldots, L. \tag{3.39}$$

The dimension of a polynomial space of order $(N_1, N_2, \ldots, N_L)$ is

$$\prod_{l=1}^{L} (N_l + 1). \tag{3.40}$$

The reproducing kernel of this space is expressed by

$$K(\boldsymbol{x}, \boldsymbol{x}') = \prod_{l=1}^{L} K_l(\xi^{(l)}, \xi^{(l)\prime}), \tag{3.41}$$

where

$$K_l(\xi^{(l)}, \xi^{(l)\prime}) = \begin{cases} \dfrac{k_N}{k_{N+1}} \dfrac{p_{N+1}(\xi^{(l)}) p_N(\xi^{(l)\prime}) - p_N(\xi^{(l)}) p_{N+1}(\xi^{(l)\prime})}{\xi^{(l)} - \xi^{(l)\prime}} & \text{if } \xi^{(l)} \neq \xi^{(l)\prime}, \\ \dfrac{k_N}{k_{N+1}} \left(p'_{N+1}(\xi^{(l)}) p_N(\xi^{(l)\prime}) - p'_N(\xi^{(l)}) p_{N+1}(\xi^{(l)\prime}) \right) & \text{if } \xi^{(l)} = \xi^{(l)\prime}. \end{cases} \tag{3.42}$$

The profile of the reproducing kernel of a polynomial space is illustrated in Figure 3.4.

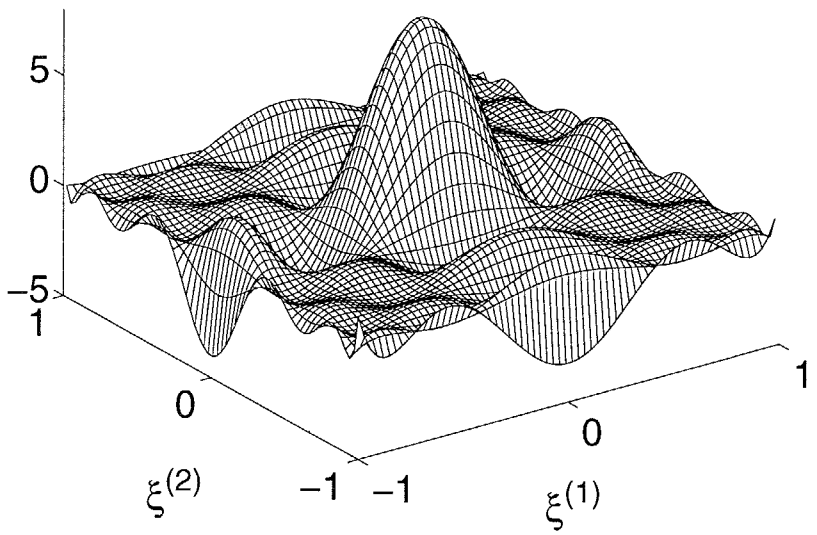


Figure 3.4: Profile of reproducing kernel $K(\boldsymbol{x}, \boldsymbol{x}')$ of a polynomial space of order $(7, 11)$ with $a_1 = a_2 = -1$, $b_1 = b_2 = 1$, $w(\boldsymbol{x}) = 1$, and $\boldsymbol{x}' = (0, 0)^\top$.

Chapter 4

Theory of model selection

4.1 Introduction

Various supervised learning methods have been developed so far, e.g. the stochastic gradient descent method (Amari [7]), the back-propagation algorithm (Rumelhart [102][103]), regularization learning (Tikhonov & Arsenin [135], Poggio & Girosi [98]), Bayesian inference (Savage [110], MacKay [68]), projection learning (Ogawa [88]), and support vector machines (Vapnik [140][141], Schölkopf *et al.* [112]). In these learning methods, the quality of learning result functions depends heavily on the choice of *models*. Here, models refer to, for example, the type and number of basis functions and parameters in learning algorithms.

If the model is too complicated, then learning result functions tend to over-fit noisy training examples. In contrast, if the model is too simple, then it is not capable of fitting training examples causing learning result functions become under-fitted. In general, both over- and under-fitted learning result functions have lower levels of the generalization capability. If the model complexity is chosen appropriately, then a good learning result function is obtained (see Figure 1.4 in page 8). Therefore, the problem of finding an appropriate model, referred to as *model selection*, is considerably important for acquiring a higher level of the generalization capability.

The goal of model selection is to obtain the optimal model that minimizes the generalization error. However, the generalization error can not be directly evaluated since the unknown learning target function is required for calculating the generalization error. A general approach to model selection is deriving an estimate of the generalization error, and then selecting the model that minimizes the estimated generalization error. In this chapter, we give an approximation of the generalization error called the *subspace*

information criterion (SIC). SIC gives an unbiased estimate of the generalization error.

4.2 Problem formulation

Suppose we are given M training examples:

$$\{(\boldsymbol{x}_m, y_m) \mid y_m = f(\boldsymbol{x}_m) + \epsilon_m\}_{m=1}^{M}. \tag{4.1}$$

Let θ be a set of factors which determine learning result functions, e.g. the type and number of basis functions, and parameters in learning algorithms. We call θ a *model*. Let $\hat{f}_\theta(\boldsymbol{x})$ be a learning result function obtained with a model θ. We assume that the learning target function $f(\boldsymbol{x})$ and the learning result function $\hat{f}_\theta(\boldsymbol{x})$ belong to a specified reproducing kernel Hilbert space H (see Section 2.3), and the generalization error of $\hat{f}_\theta(\boldsymbol{x})$ is measured by

$$J_G[\theta] = \mathrm{E}_{\boldsymbol{\epsilon}} \|\hat{f}_\theta - f\|^2, \tag{4.2}$$

where $\mathrm{E}_{\boldsymbol{\epsilon}}$ denotes the expectation over the noise. The norm is typically defined as

$$\|\hat{f}_\theta - f\|^2 = \int \left|\hat{f}_\theta(\boldsymbol{u}) - f(\boldsymbol{u})\right|^2 w(\boldsymbol{u}) d\boldsymbol{u}, \tag{4.3}$$

where the integral with respect to $\boldsymbol{u}$ means the expectation over future sample points $\boldsymbol{u}$ and $w(\boldsymbol{u})$ is some weight function, e.g., the probability density function of $\boldsymbol{u}$. Then the problem of model selection considered in this chapter is formulated as follows.

Definition 4.1 (Model selection) *From a set $\mathcal{M}$ of model candidates, select the best model $\hat{\theta}$ that minimizes the generalization error J_G:*

$$\hat{\theta} = \operatorname*{argmin}_{\theta \in \mathcal{M}} J_G[\theta]. \tag{4.4}$$

4.3 Subspace information criterion

In this section, we derive an approximation of the generalization error J_G called the *subspace information criterion (SIC)*.

4.3.1 Setting

In the derivation of SIC, we assume the following conditions.

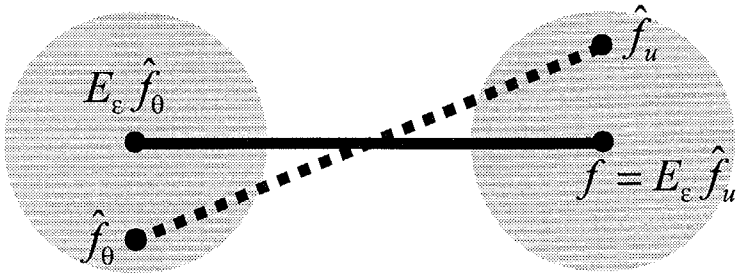


Figure 4.1: Basic idea of SIC. The solid line denotes the bias of $\hat{f}_\theta$. It can be roughly estimated by the dotted line, which can be calculated.

1. The learning result function $\hat{f}_\theta(\boldsymbol{x})$ obtained with the model θ is given by using a linear operator X_θ as

$$\hat{f}_\theta = X_\theta \boldsymbol{y}. \tag{4.5}$$

The vector $\boldsymbol{y}$ is defined as

$$\boldsymbol{y} = (y_1, y_2, \ldots, y_M)^\top, \tag{4.6}$$

where $\top$ denotes the transpose of a vector.

2. A linear operator X_u which gives an unbiased learning result function $\hat{f}_u(\boldsymbol{x})$ is available:

$$\mathrm{E}_{\boldsymbol{\epsilon}} \hat{f}_u = f, \tag{4.7}$$

where

$$\hat{f}_u = X_u \boldsymbol{y}. \tag{4.8}$$

3. The mean noise is zero:

$$\mathrm{E}_{\boldsymbol{\epsilon}} \boldsymbol{\epsilon} = 0, \tag{4.9}$$

where

$$\boldsymbol{\epsilon} = (\epsilon_1, \epsilon_2, \ldots, \epsilon_M)^\top. \tag{4.10}$$

Assumption 1 implies that the range of X_θ becomes a subspace of H. This is the origin of the name *subspace information criterion*. The main idea of SIC is that the unbiased learning result function $\hat{f}_u(\boldsymbol{x})$ is used for estimating the generalization error of $\hat{f}_\theta(\boldsymbol{x})$ (Figure 4.1).

4.3.2 Derivation of SIC

It follows from Eqs.(4.1), (4.6), and (4.10) that

$$\boldsymbol{y} = \boldsymbol{z} + \boldsymbol{\epsilon}, \tag{4.11}$$

where

$$\boldsymbol{z} = (f(\boldsymbol{x}_1), f(\boldsymbol{x}_2), \ldots, f(\boldsymbol{x}_M))^\top. \tag{4.12}$$

Let Q be the noise covariance matrix:

$$Q = \mathrm{E}_{\boldsymbol{\epsilon}} \left(\boldsymbol{\epsilon} \otimes \overline{\boldsymbol{\epsilon}} \right). \tag{4.13}$$

It is known that the generalization error of $\hat{f}_\theta(\boldsymbol{x})$ can be decomposed into the *bias* and *variance* (see e.g. Takemura [132], Geman *et al.* [40], Efron & Tibshirani [33]):

$$J_G[\theta] = \|\mathrm{E}_{\boldsymbol{\epsilon}} \hat{f}_\theta - f\|^2 + \mathrm{E}_{\boldsymbol{\epsilon}} \|\hat{f}_\theta - \mathrm{E}_{\boldsymbol{\epsilon}} \hat{f}_\theta\|^2. \tag{4.14}$$

Then it follows from Eqs.(4.14), (4.5), (4.7), (4.11), (4.8), (4.9), and (4.13) that $J_G[\theta]$ can be exactly expressed by using $\hat{f}_u(\boldsymbol{x})$ and Q as

$$\begin{aligned}
J_G[\theta] &= \|\hat{f}_\theta - \hat{f}_u\|^2 - \|\hat{f}_\theta - \hat{f}_u\|^2 + \|\mathrm{E}_{\boldsymbol{\epsilon}} \hat{f}_\theta - f\|^2 + \mathrm{E}_{\boldsymbol{\epsilon}} \|X_\theta \boldsymbol{y} - \mathrm{E}_{\boldsymbol{\epsilon}} X_\theta \boldsymbol{y}\|^2 \\
&= \|\hat{f}_\theta - \hat{f}_u\|^2 - \|\mathrm{E}_{\boldsymbol{\epsilon}}(\hat{f}_\theta - \hat{f}_u) - \mathrm{E}_{\boldsymbol{\epsilon}}(\hat{f}_\theta - \hat{f}_u) + \hat{f}_\theta - \hat{f}_u\|^2 \\
&\quad + \|\mathrm{E}_{\boldsymbol{\epsilon}}(\hat{f}_\theta - \hat{f}_u)\|^2 + \mathrm{E}_{\boldsymbol{\epsilon}} \|X_\theta(\boldsymbol{z} + \boldsymbol{\epsilon}) - \mathrm{E}_{\boldsymbol{\epsilon}} X_\theta(\boldsymbol{z} + \boldsymbol{\epsilon})\|^2 \\
&= \|(X_\theta - X_u)\boldsymbol{y}\|^2 - \|\mathrm{E}_{\boldsymbol{\epsilon}}(\hat{f}_\theta - \hat{f}_u)\|^2 - 2\mathrm{Re}\langle \mathrm{E}_{\boldsymbol{\epsilon}}(\hat{f}_\theta - \hat{f}_u), -\mathrm{E}_{\boldsymbol{\epsilon}}(\hat{f}_\theta - \hat{f}_u) + \hat{f}_\theta - \hat{f}_u \rangle \\
&\quad - \|\mathrm{E}_{\boldsymbol{\epsilon}}(\hat{f}_\theta - \hat{f}_u) - (\hat{f}_\theta - \hat{f}_u)\|^2 + \|\mathrm{E}_{\boldsymbol{\epsilon}}(\hat{f}_\theta - \hat{f}_u)\|^2 + \mathrm{E}_{\boldsymbol{\epsilon}} \|X_\theta \boldsymbol{\epsilon}\|^2 \\
&= \|(X_\theta - X_u)\boldsymbol{y}\|^2 - 2\mathrm{Re}\langle \mathrm{E}_{\boldsymbol{\epsilon}}(\hat{f}_\theta - \hat{f}_u), -\mathrm{E}_{\boldsymbol{\epsilon}}(\hat{f}_\theta - \hat{f}_u) + \hat{f}_\theta - \hat{f}_u \rangle \\
&\quad - \|\mathrm{E}_{\boldsymbol{\epsilon}}(\hat{f}_\theta - \hat{f}_u) - (\hat{f}_\theta - \hat{f}_u)\|^2 + \mathrm{tr} X_\theta Q X_\theta^*,
\end{aligned} \tag{4.15}$$

where 'Re' stands for the real part of a complex number. The second and third terms in Eq.(4.15) can not be directly evaluated since $\mathrm{E}_{\boldsymbol{\epsilon}}(\hat{f}_\theta - \hat{f}_u)$ is unknown, so we shall average these terms out over the noise. Then the second term vanishes:

$$\mathrm{E}_{\boldsymbol{\epsilon}} \left(-2\mathrm{Re}\langle \mathrm{E}_{\boldsymbol{\epsilon}}(\hat{f}_\theta - \hat{f}_u), -\mathrm{E}_{\boldsymbol{\epsilon}}(\hat{f}_\theta - \hat{f}_u) + \hat{f}_\theta - \hat{f}_u \rangle \right) = 0. \tag{4.16}$$

And it follows from Eqs.(4.5), (4.8), (4.11), (4.9), and (4.13) that the third term yields

$$\mathrm{E}_{\boldsymbol{\epsilon}} \left(-\|\mathrm{E}_{\boldsymbol{\epsilon}}(\hat{f}_\theta - \hat{f}_u) - (\hat{f}_\theta - \hat{f}_u)\|^2 \right)$$

$$
\begin{aligned}
&= -\mathrm{E}_{\boldsymbol{\epsilon}}\|\mathrm{E}_{\boldsymbol{\epsilon}}(X_\theta - X_u)\boldsymbol{y} - (X_\theta - X_u)\boldsymbol{y}\|^2 \\
&= -\mathrm{E}_{\boldsymbol{\epsilon}}\|\mathrm{E}_{\boldsymbol{\epsilon}}(X_\theta - X_u)(\boldsymbol{z}+\boldsymbol{\epsilon}) - (X_\theta - X_u)(\boldsymbol{z}+\boldsymbol{\epsilon})\|^2 \\
&= -\mathrm{E}_{\boldsymbol{\epsilon}}\|(X_\theta - X_u)\boldsymbol{\epsilon}\|^2 \\
&= -\mathrm{tr}(X_\theta - X_u)Q(X_\theta - X_u)^*. \qquad (4.17)
\end{aligned}
$$

Then we have the following criterion.

Definition 4.2 (Subspace information criterion) *The following functional* SIC[θ] *is called the subspace information criterion for a model* θ.

$$\mathrm{SIC}[\theta] = \|(X_\theta - X_u)\boldsymbol{y}\|^2 - \mathrm{tr}(X_\theta - X_u)Q(X_\theta - X_u)^* + \mathrm{tr}X_\theta Q X_\theta^*. \qquad (4.18)$$

The model that minimizes SIC is called the *minimum SIC (MSIC) model.* The effectiveness of SIC as a model selection criterion is theoretically substantiated by the following lemma.

Lemma 4.3 *SIC is an unbiased estimate of the generalization error* J_G*:*

$$\mathrm{E}_{\boldsymbol{\epsilon}}\mathrm{SIC}[\theta] = J_G[\theta]. \qquad (4.19)$$

A proof of Lemma 4.3 is given in Section 4.9.1.

Since the bias is always non-negative from the definition, we can also consider the following corrected SIC (cSIC):

$$\mathrm{cSIC}[\theta] = \max\left(0, \|(X_\theta - X_u)\boldsymbol{y}\|^2 - \mathrm{tr}(X_\theta - X_u)Q(X_\theta - X_u)^*\right) + \mathrm{tr}X_\theta Q X_\theta^*. \qquad (4.20)$$

4.3.3 Practical expression of SIC

Although SIC defined by Eq.(4.18) does not include the unknown learning target function $f(\boldsymbol{x})$, it still includes factors which are often unknown, e.g. an operator X_u that gives an unbiased learning result function $\hat{f}_u(\boldsymbol{x})$ and the noise covariance matrix Q. Here, we show their practical estimation methods by further assuming the following conditions:

1. The function space H to which the learning target function $f(\boldsymbol{x})$ belongs is finite dimensional:

$$\dim H < \infty. \qquad (4.21)$$

2. The norm in H is computable. For example, when the norm is expressed as Eq.(4.3), the covariance operator V of the weight function $w(\boldsymbol{u})$ is assumed to be known:

$$V = \int \left(K(\cdot, \boldsymbol{u}) \otimes \overline{K(\cdot, \boldsymbol{u})} \right) w(\boldsymbol{u}) d\boldsymbol{u}, \tag{4.22}$$

where $K(\cdot, \cdot)$ is the reproducing kernel of H (see Section 2.3).

3. The functions $\{K(\boldsymbol{x}, \boldsymbol{x}_m)\}_{m=1}^M$ span the whole space H:

$$\mathcal{L}\left(\{K(\boldsymbol{x}, \boldsymbol{x}_m)\}_{m=1}^M\right) = H, \tag{4.23}$$

where $\mathcal{L}\left(\{K(\boldsymbol{x}, \boldsymbol{x}_m)\}_{m=1}^M\right)$ denotes the linear manifold spanned by $\{K(\boldsymbol{x}, \boldsymbol{x}_m)\}_{m=1}^M$.

4. The number M of training examples is larger than the dimension of the function space H to which the learning target function $f(\boldsymbol{x})$ belongs:

$$M > \dim H. \tag{4.24}$$

Let A be an operator defined as

$$A = \sum_{m=1}^{M} \left(\boldsymbol{e}_m \otimes \overline{K(\cdot, \boldsymbol{x}_m)} \right), \tag{4.25}$$

where $(\cdot \otimes \bar{\cdot})$ denotes the Neumann-Schatten product (see Section 2.2.4) and $\boldsymbol{e}_m$ is the m-th vector of the so-called standard basis in $\mathbf{C}^M$. Note that it holds for any function f in H that

$$Af = \boldsymbol{z}, \tag{4.26}$$

where $\boldsymbol{z}$ is the vector defined by Eq.(4.12). This can be verified from the property of the reproducing kernel (see Section 2.3):

$$\langle f(\cdot), K(\cdot, \boldsymbol{x}') \rangle = f(\boldsymbol{x}'). \tag{4.27}$$

Note that Eq.(4.23) is equivalently expressed as

$$\mathcal{R}(A^*) = H, \tag{4.28}$$

where A^* is the adjoint operator of A. Then it follows from Eqs.(4.11), (4.26), (4.9), and (4.28) that

$$\begin{aligned} \mathrm{E}_{\boldsymbol{\epsilon}} A^\dagger \boldsymbol{y} &= \mathrm{E}_{\boldsymbol{\epsilon}} A^\dagger (\boldsymbol{z} + \boldsymbol{\epsilon}) = A^\dagger \boldsymbol{z} + \mathrm{E}_{\boldsymbol{\epsilon}} A^\dagger \boldsymbol{\epsilon} \\ &= A^\dagger A f = P_{\mathcal{R}(A^*)} f = I_H f \\ &= f, \end{aligned} \tag{4.29}$$

where $P_{\mathcal{R}(A^*)}$ denotes the orthogonal projection operator onto the range of A^* and I_H denotes the identity operator on H. This implies that $A^\dagger \boldsymbol{y}$ is an unbiased estimate of f. Therefore, $A^\dagger$ can be used as X_u:

$$X_u = A^\dagger. \tag{4.30}$$

The noise covariance matrix Q can be estimated as (see e.g. Fedorov [34])

$$\hat{Q} = \hat{\sigma}^2 I_M, \quad \hat{\sigma}^2 = \frac{\sum_{m=1}^{M} \left| \hat{f}_u(\boldsymbol{x}_m) - y_m \right|^2}{M - \dim H}, \tag{4.31}$$

where I_M is the M-dimensional identity matrix. If the noise covariance matrix Q is given as $Q = \sigma^2 I_M$ with $\sigma^2 > 0$, then $\hat{\sigma}^2$ is an unbiased estimate of σ^2 (see Proposition 4.14 in page 44). Therefore, SIC with Q estimated by Eq.(4.31) still gives an unbiased estimate of the generalization error, i.e., Lemma 4.3 holds.

SIC with X_u obtained by Eq.(4.30) and Q estimated by Eq.(4.31) is given as

$$\mathrm{SIC}[\theta] = \|(X_\theta - A^\dagger)\boldsymbol{y}\|^2 - \hat{\sigma}^2 \mathrm{tr}(X_\theta - A^\dagger)(X_\theta - A^\dagger)^* + \hat{\sigma}^2 \mathrm{tr} X_\theta X_\theta^*. \tag{4.32}$$

4.4 SIC for least mean squares learning

In this section, SIC is applied to least mean squares (LMS) learning.

4.4.1 Least mean squares learning

LMS learning gives a learning result function $\hat{f}_S(\boldsymbol{x})$ that minimizes the training error in a reproducing kernel Hilbert space S included in H (see Section 3.2.1):

$$\hat{f}_S = \operatorname*{argmin}_{\hat{f} \in S} J_{TE}[\hat{f}], \tag{4.33}$$

where

$$J_{TE}[\hat{f}] = \frac{1}{M} \sum_{m=1}^{M} \left| \hat{f}(\boldsymbol{x}_m) - y_m \right|^2. \tag{4.34}$$

Let us denote the reproducing kernel of S by $K_S(\boldsymbol{x}, \boldsymbol{x}')$, and let A_S be an operator from H to $\mathbf{C}^M$ defined as

$$A_S = \sum_{m=1}^{M} \left(\boldsymbol{e}_m \otimes \overline{K_S(\cdot, \boldsymbol{x}_m)} \right), \tag{4.35}$$

where $(\cdot \otimes \bar{\cdot})$ denotes the Neumann-Schatten product (see Section 2.2.4) and $\boldsymbol{e}_m$ is the m-th vector of the so-called standard basis in $\mathbf{C}^M$. Then the LMS learning function $\hat{f}_S(x)$

is given as

$$\hat{f}_S = X_S \boldsymbol{y}, \tag{4.36}$$

where

$$X_S = A_S^\dagger. \tag{4.37}$$

It is known that the selection of the subspace S is crucial for acquiring a higher level of the generalization capability. Here, we shall discuss the problem of determining the subspace S for the optimal generalization capability. As shown in Lemma 4.3, SIC gives an unbiased estimate of the generalization error J_G. Therefore, we will use SIC as a substitute for the generalization error. From Eqs.(4.32) and (4.37), SIC for LMS learning with X_u obtained by Eq.(4.30) and Q estimated by Eq.(4.31) is given as

$$\mathrm{SIC}[S] = \|(A_S^\dagger - A^\dagger)\boldsymbol{y}\|^2 - \hat{\sigma}^2 \mathrm{tr}(A_S^\dagger - A^\dagger)(A_S^\dagger - A^\dagger)^* + \hat{\sigma}^2 \mathrm{tr} A_S^\dagger (A_S^\dagger)^*. \tag{4.38}$$

4.4.2 Optimal selection of subspace models from given candidates

If a set $\mathcal{M}$ of a finite number of subspaces is given as model candidates, SIC can be used for selecting the optimal subspace from the given set $\mathcal{M}$, i.e., calculate SIC for each model in $\mathcal{M}$ and select the model that minimizes SIC. Here, we show a practical calculation method of SIC and LMS learning function $\hat{f}_S(\boldsymbol{x})$ by matrix operations.

Let H be spanned by μ linearly independent (generally non-orthogonal) functions $\{\varphi_p(\boldsymbol{x})\}_{p=1}^{\mu}$. Let us define the inner product in H as

$$\langle f, g \rangle = \int f(\boldsymbol{u}) \overline{g(\boldsymbol{u})} p(\boldsymbol{u}) d\boldsymbol{u}, \tag{4.39}$$

where $\overline{\cdot}$ denotes the complex conjugate of a scalar and $p(\boldsymbol{u})$ is the probability density function of future sample points $\boldsymbol{u}$. Let B be an $M \times \mu$ matrix with the (m,p)-th element being $\varphi_p(\boldsymbol{x}_m)$:

$$[B]_{m,p} = \varphi_p(\boldsymbol{x}_m), \tag{4.40}$$

where $[\,\cdot\,]_{m,p}$ denotes the (m,p)-th element of a matrix. The matrix B is called the *design matrix* (see e.g. Efron & Tibshirani [33]). Let S be spanned by a subset of $\{\varphi_p(\boldsymbol{x})\}_{p=1}^{\mu}$. Let B_S be an $M \times \mu$ matrix with the (m,p)-th element being

$$[B_S]_{m,p} = \begin{cases} \varphi_p(\boldsymbol{x}_m) & \text{if } \varphi_p \in S, \\ 0 & \text{otherwise.} \end{cases} \tag{4.41}$$

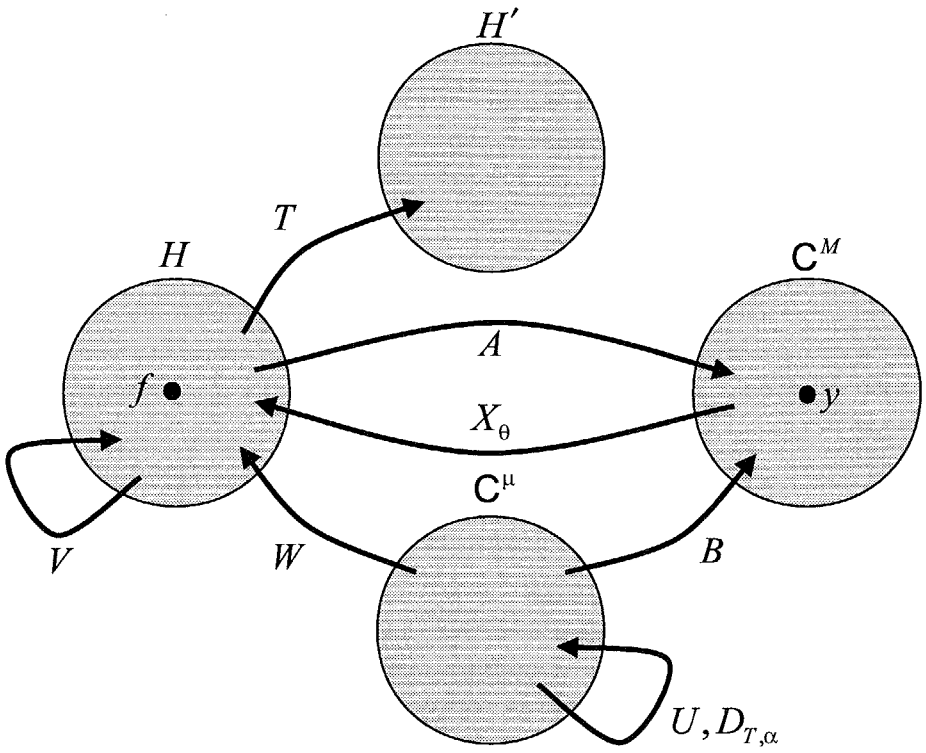


Figure 4.2: Matrices and operators.

Let W be an operator from $\mathbf{C}^{\mu}$ to H defined as

$$W = \sum_{p=1}^{\mu} (\varphi_p \otimes \overline{\boldsymbol{e}_p}), \tag{4.42}$$

where $\boldsymbol{e}_p$ is the p-th vector of the so-called standard basis in $\mathbf{C}^{\mu}$. Let U be a μ-dimensional matrix with the (p, p')-th element being $\langle V\varphi_{p'}, \varphi_p \rangle$:

$$[U]_{p,p'} = \langle V\varphi_{p'}, \varphi_p \rangle, \tag{4.43}$$

where the operator V is defined by Eq.(4.22), which is assumed to be known (see Section 4.3.3). Note that $[U]_{p,p'}$ is expressed as

$$[U]_{p,p'} = \int \varphi_{p'}(\boldsymbol{u})\overline{\varphi_p(\boldsymbol{u})}p(\boldsymbol{u})d\boldsymbol{u}. \tag{4.44}$$

Therefore, when $\{\varphi_p(\boldsymbol{x})\}_{p=1}^{\mu}$ is orthonormal, the matrix U is reduced to the identity matrix. The above matrices and operators are summarized in Figure 4.2. Then we have the following corollaries.

Corollary 4.4 (Calculation of SIC for LMS learning) *SIC given by Eq.(4.38) can be calculated as*

$$\begin{aligned}\mathrm{SIC}[S] = &\langle U(B_S^\dagger - B^\dagger)\boldsymbol{y}, (B_S^\dagger - B^\dagger)\boldsymbol{y}\rangle + 2\hat{\sigma}^2 \mathrm{Re}\ \mathrm{tr} U B_S^\dagger (B^\dagger)^* \\ &- \hat{\sigma}^2 \mathrm{tr} U B^\dagger (B^\dagger)^*, \end{aligned} \tag{4.45}$$

where

$$\hat{\sigma}^2 = \frac{\langle \boldsymbol{y} - BB^\dagger \boldsymbol{y}, \boldsymbol{y}\rangle}{M - \mu}. \tag{4.46}$$

Corollary 4.5 (Calculation of LMS learning functions) *The LMS learning function $\hat{f}_S(\boldsymbol{x})$ for a model S can be calculated as*

$$\hat{f}_S(\boldsymbol{x}) = \sum_{p:\varphi_p \in S} [B_S^\dagger \boldsymbol{y}]_p \varphi_p(\boldsymbol{x}), \tag{4.47}$$

where $[\,\cdot\,]_p$ denotes the p-th element of a vector.

Proofs of Corollaries 4.4 and 4.5 are provided in Sections 4.9.2 and 4.9.3, respectively.

If terms which are irrelevant to the model S are ignored, Eq.(4.45) is reduced to

$$\langle U B_S^\dagger \boldsymbol{y}, B_S^\dagger \boldsymbol{y}\rangle - 2\mathrm{Re}\ \langle U B_S^\dagger \boldsymbol{y}, B^\dagger \boldsymbol{y}\rangle + 2\hat{\sigma}^2 \mathrm{Re}\ \mathrm{tr} U B_S^\dagger (B^\dagger)^*. \tag{4.48}$$

In practice, the calculation of the Moore-Penrose generalized inverse is sometimes unstable. To overcome the unstableness, we recommend using *Tikhonov's regularization* (Tikhonov & Arsenin [135]):

$$B_S^\dagger \longleftarrow (B_S^* B_S + \gamma I_\mu)^{-1} B_S^*, \tag{4.49}$$

$$B^\dagger \longleftarrow (B^* B + \gamma I_\mu)^{-1} B^*, \tag{4.50}$$

where γ is a small positive constant and I_μ is the μ-dimensional identity matrix.

Using Eq.(4.45), we can select the optimal subspace from a set of finite candidates.

4.5 SIC for regularization learning

In this section, SIC is applied to regularization learning.

4.5.1 Regularization learning

Regularization learning gives a learning result function $\hat{f}_{T,\alpha}(\boldsymbol{x})$ that minimizes the following J_R in H (see Section 3.2.2):

$$\hat{f}_{T,\alpha} = \underset{\hat{f}\in H}{\operatorname{argmin}} J_R[\hat{f}], \tag{4.51}$$

where

$$J_R[\hat{f}] = \sum_{m=1}^{M} \left|\hat{f}(\boldsymbol{x}_m) - y_m\right|^2 + \alpha\|T\hat{f}\|^2. \tag{4.52}$$

T and α are called the regularization operator and regularization parameter, respectively. The regularization learning function $\hat{f}_{T,\alpha}(\boldsymbol{x})$ for the regularization operator T and regularization parameter α is given as

$$\hat{f}_{T,\alpha} = X_{T,\alpha}\boldsymbol{y}. \tag{4.53}$$

Here, $X_{T,\alpha}$ is defined as

$$X_{T,\alpha} = (A^*A + \alpha T^*T)^\dagger A^*, \tag{4.54}$$

where A is defined by Eq.(4.25).

It is known that the selection of the regularization operator T and regularization parameter α is crucial for acquiring a higher level of the generalization capability. Here, we shall discuss the problem of determining T and α for the optimal generalization capability. As shown in Lemma 4.3, SIC gives an unbiased estimate of the generalization error J_G. Therefore, we will use SIC as a substitute for the generalization error. From Eqs.(4.32) and (4.54), SIC for regularization learning with X_u obtained by Eq.(4.30) and Q estimated by Eq.(4.31) is expressed as

$$\mathrm{SIC}[T,\alpha] = \|(X_{T,\alpha} - A^\dagger)\boldsymbol{y}\|^2 - \hat{\sigma}^2\mathrm{tr}(X_{T,\alpha} - A^\dagger)(X_{T,\alpha} - A^\dagger)^* + \hat{\sigma}^2\mathrm{tr}X_{T,\alpha}X_{T,\alpha}^*. \tag{4.55}$$

In Section 4.5.2, SIC is used for selecting the optimal regularization operator T and regularization parameter α from given, finite candidates. This method can be applied to any situations. In contrast, Section 4.5.3 considers certain conditions and give the closed form of the optimal regularization parameter α that minimizes SIC. This corresponds to selecting the optimal regularization parameter from an infinite number of candidates.

4.5.2 Optimal selection of regularization operator and regularization parameter from given candidates

If a set $\{(T, \alpha)\}$ of finite pairs of the regularization operator and regularization parameter is given as candidates, SIC can be used for selecting the optimal pair from the given set, i.e., calculate SIC for each pair (T, α) and select the minimizer. Here, we show a practical calculation method of SIC and the regularization learning function $\hat{f}_{T,\alpha}(\boldsymbol{x})$ for a given pair (T, α) by matrix operations.

Let H be spanned by μ linearly independent (generally non-orthogonal) functions $\{\varphi_p(\boldsymbol{x})\}_{p=1}^{\mu}$. Let us define the inner product in H by Eq.(4.39). Let $D_{T,\alpha}$ be a μ-dimensional matrix defined as

$$D_{T,\alpha} = B^*B + \alpha W^*T^*TW, \tag{4.56}$$

where B and W are defined by Eqs.(4.40) and (4.42), respectively. Note that W^*T^*TW is a μ-dimensional matrix, and $D_{T,\alpha}$ is non-singular. The above matrices and operators are summarized in Figure 4.2 in page 37. Then the following corollaries hold.

Corollary 4.6 (Calculation of SIC for regularization learning) *SIC given by Eq.(4.55) can be calculated as*

$$\begin{aligned}\mathrm{SIC}[T, \alpha] = &\langle U(D_{T,\alpha}^{-1}B^* - B^\dagger)\boldsymbol{y}, (D_{T,\alpha}^{-1}B^* - B^\dagger)\boldsymbol{y}\rangle + 2\hat{\sigma}^2 \mathrm{tr} U D_{T,\alpha}^{-1} \\ &- \hat{\sigma}^2 \mathrm{tr} U B^\dagger (B^\dagger)^*,\end{aligned} \tag{4.57}$$

where U is given by Eq.(4.44) and $\hat{\sigma}^2$ is given by Eq.(4.46).

Corollary 4.7 (Calculation of regularization learning function) *The regularization learning function $\hat{f}_{T,\alpha}(\boldsymbol{x})$ for (T, α) can be calculated as*

$$\hat{f}_{T,\alpha}(\boldsymbol{x}) = \sum_{p=1}^{\mu} [D_{T,\alpha}^{-1}B^*\boldsymbol{y}]_p \varphi_p(\boldsymbol{x}), \tag{4.58}$$

where $[\,\cdot\,]_p$ denotes the p-th element of a vector.

Proofs of Corollaries 4.6 and 4.7 are provided in Sections 4.9.4 and 4.9.5, respectively.

If terms which are irrelevant to T and α are ignored, then Eq.(4.57) is reduced to

$$\langle U D_{T,\alpha}^{-1}B^*\boldsymbol{y}, D_{T,\alpha}^{-1}B^*\boldsymbol{y}\rangle - 2\mathrm{Re}\langle U D_{T,\alpha}^{-1}B^*\boldsymbol{y}, B^\dagger\boldsymbol{y}\rangle + 2\hat{\sigma}^2 \mathrm{tr} U D_{T,\alpha}^{-1}. \tag{4.59}$$

Using Eq.(4.57), we can select the optimal regularization operator T and regularization parameter α from a set $\{(T, \alpha)\}$ of finite candidates.

4.5.3 Active design of optimal regularization parameter

The design method of the regularization operator T and regularization parameter α described above is rather passive since they are selected from given, finite candidates. Here, we give a method of actively determining the optimal regularization parameter that minimizes SIC.

4.5.3.1 Second order approximation

We adopt the identity operator on H as the regularization operator:

$$T = I_H. \tag{4.60}$$

Let us denote the regularization learning operator with $T = I_H$ by X_α. Then X_α is given as

$$X_\alpha = X_{I_H,\alpha} = (A^*A + \alpha I_H)^{-1}A^*, \tag{4.61}$$

and SIC with $T = I_H$ is expressed as

$$\mathrm{SIC}[\alpha] = \|(X_\alpha - A^\dagger)\boldsymbol{y}\|^2 - \hat{\sigma}^2\mathrm{tr}(X_\alpha - A^\dagger)(X_\alpha - A^\dagger)^* + \hat{\sigma}^2\mathrm{tr}X_\alpha X_\alpha^*. \tag{4.62}$$

First, we give another expression of the regularization learning operator X_α.

Lemma 4.8 *Under assumptions (4.23) and (4.60), the regularization learning operator X_α is expressed as*

$$\begin{aligned} X_\alpha = &\sum_{j=1}^{n}(-\alpha)^{j-1}(A^*A)^{-j}A^* \\ &+ (-\alpha)^n(A^*A)^{-(n+1)}(I_H + \alpha(A^*A)^{-1})^{-1}A^*, \end{aligned} \tag{4.63}$$

where n is an arbitrary fixed positive integer.

A proof of Lemma 4.8 is given in Section 4.9.6.

Let $\lambda_{\max}$ be the maximum eigenvalue of $(A^*A)^{-1}$. Then we have the following lemma.

Lemma 4.9 $\mathrm{SIC}[\alpha]$ *given by Eq.(4.62) is approximated as*

$$\begin{aligned} \widehat{\mathrm{SIC}}[\alpha] = &\alpha^2\left(\|(A^*A)^{-2}A^*\boldsymbol{y}\|^2 + 2\hat{\sigma}^2\mathrm{tr}(A^*A)^{-3}\right) \\ &- 2\alpha\hat{\sigma}^2\mathrm{tr}(A^*A)^{-2} + \hat{\sigma}^2\mathrm{tr}(A^*A)^{-1} \end{aligned} \tag{4.64}$$

with precision $\mathcal{O}\left((\lambda_{\max}\alpha)^3\right)$:

$$\widehat{\mathrm{SIC}}[\alpha] - \mathrm{SIC}[\alpha] = \mathcal{O}\left((\lambda_{\max}\alpha)^3\right). \tag{4.65}$$

A proof of Lemma 4.9 is given in Section 4.9.7.

Let α_{SIC} and $\alpha_{\widehat{SIC}}$ be the minimizers of SIC and $\widehat{\mathrm{SIC}}$, respectively:

$$\alpha_{SIC} = \operatorname*{argmin}_{\alpha} \mathrm{SIC}[\alpha], \tag{4.66}$$

$$\alpha_{\widehat{SIC}} = \operatorname*{argmin}_{\alpha} \widehat{\mathrm{SIC}}[\alpha]. \tag{4.67}$$

Then we have the following theorem.

Theorem 4.10 *Under assumptions (4.23), (4.24), and (4.60), $\alpha_{\widehat{SIC}}$ is given as*

$$\alpha_{\widehat{SIC}} = \frac{\hat{\sigma}^2 \mathrm{tr}(A^*A)^{-2}}{\|(A^*A)^{-2}A^*\boldsymbol{y}\|^2 + 2\hat{\sigma}^2 \mathrm{tr}(A^*A)^{-3}}, \tag{4.68}$$

and the following relation holds:

$$\mathrm{SIC}[\alpha_{\widehat{SIC}}] - \mathrm{SIC}[\alpha_{SIC}] = \mathcal{O}\left((\lambda_{\max}\alpha_{\widehat{SIC}})^3 + (\lambda_{\max}\alpha_{SIC})^3\right). \tag{4.69}$$

A proof of Theorem 4.10 is provided in Section 4.9.8.

Lemma 4.9 and Theorem 4.10 mean that if $(\lambda_{\max}\alpha_{\widehat{SIC}})^3$ and $(\lambda_{\max}\alpha_{SIC})^3$ are small enough to be neglected, then $\alpha_{\widehat{SIC}}$ almost minimizes SIC. In practice, it is possible to calculate $\alpha_{\widehat{SIC}}$ and $\lambda_{\max}$ from given training examples. However, α_{SIC} can not be directly evaluated. Further studies will be needed for assessing the validity of $\alpha_{\widehat{SIC}}$ (see Section 4.8.2.3 for experimental evaluation).

Finally, we give a matrix expression of $\alpha_{\widehat{SIC}}$.

Corollary 4.11 *$\alpha_{\widehat{SIC}}$ given by Eq.(4.68) is expressed as*

$$\alpha_{\widehat{SIC}} = \frac{\hat{\sigma}^2 \mathrm{tr}(UY)^2}{\langle (UY)^2 UB^\dagger \boldsymbol{y}, B^\dagger \boldsymbol{y}\rangle + 2\hat{\sigma}^2 \mathrm{tr}(UY)^3}, \tag{4.70}$$

where U, B, and $\hat{\sigma}^2$ are given by Eqs.(4.44), (4.40), and (4.46), respectively, and Y is defined as

$$Y = B^\dagger (B^\dagger)^*. \tag{4.71}$$

A proof of Corollary 4.11 is given in Section 4.9.9.

Note that similar discussion is possible for $T = W^{-1}$, i.e., weight decay. In this case, we have

$$\alpha_{\widehat{SIC}} = \frac{\hat{\sigma}^2 \mathrm{tr}UY^2}{\langle YUYB^\dagger \boldsymbol{y}, B^\dagger \boldsymbol{y}\rangle + 2\hat{\sigma}^2 \mathrm{tr}UY^3}. \tag{4.72}$$

4.5.3.2 When $\frac{1}{M}A^*A = I_H$

We gave the closed form of the regularization parameter that minimizes SIC when it is approximated up to the second order terms. Here, we derive the rigorous solution by further assuming the following condition:

$$\frac{1}{M}A^*A = I_H. \tag{4.73}$$

Note that Eq.(4.73) exactly holds if H is a trigonometric polynomial space and sample points $\{\boldsymbol{x}_m\}_{m=1}^M$ are fixed to regular intervals in the domain (see Section 5.3.4 for detail).

Under assumptions (4.60) and (4.73), it follows from Eq.(4.61) that the regularization operator X_α is given as

$$X_\alpha = (MI_H + \alpha I_H)^{-1}A^* = \frac{1}{M+\alpha}A^*. \tag{4.74}$$

Then the closed form of the regularization parameter that minimizes SIC is given as follows.

Theorem 4.12 *Under assumptions (4.24), (4.60), (4.73), and*

$$\frac{1}{M}\sum_{m=1}^{M}|y_m|^2 > \hat{\sigma}^2 > 0, \tag{4.75}$$

the regularization parameter α_{SIC} that minimizes SIC is given as

$$\alpha_{SIC} = \frac{\hat{\sigma}^2\mu}{\frac{1}{M}\sum_{m=1}^{M}|y_m|^2 - \hat{\sigma}^2}, \tag{4.76}$$

where $\hat{\sigma}^2$ is given by Eq.(4.31) and μ is the dimension of H.

A proof of Theorem 4.12 is given in Section 4.9.10.

Eq.(4.75) implies that the sampled signal level is larger than the noise level. If Eq.(4.75) does not holds, then α_{SIC} tends to be ∞.

Now we assume that the noise covariance matrix Q is given as

$$Q = \sigma^2 I_M \tag{4.77}$$

with $\sigma^2 > 0$, where I_M is the M-dimensional identity matrix. Then the following propositions and lemma show the validity of α_{SIC}.

Proposition 4.13 *(Hagiwara & Kuno [46]) Under assumptions (4.60), (4.73), (4.77), and $\|f\| > 0$, the optimal regularization parameter α_{OPT} that minimizes the generalization error J_G is given as*

$$\alpha_{OPT} = \frac{\sigma^2 \mu}{\|f\|^2}. \tag{4.78}$$

Proposition 4.14 *(Fedorov [34]) Under assumptions (4.23), (4.24), and (4.77), it holds that*

$$\mathrm{E}_{\boldsymbol{\epsilon}} \hat{\sigma}^2 = \sigma^2. \tag{4.79}$$

Lemma 4.15 *Under assumptions (4.24), (4.73), and (4.77), it holds that*

$$\mathrm{E}_{\boldsymbol{\epsilon}} \left(\frac{1}{M} \sum_{m=1}^{M} |y_m|^2 - \hat{\sigma}^2 \right) = \|f\|^2. \tag{4.80}$$

A proof of Lemma 4.15 is given in Section 4.9.11.

Eqs.(4.79) and (4.80) imply that α_{SIC} given by Eq.(4.76) can be regarded as an estimate of α_{OPT} with the denominator and numerator in Eq.(4.78) estimated by their unbiased estimates. This assures the validity of α_{SIC}.

4.6 Comparison with existing model selection techniques

In this section, SIC is theoretically compared with existing model selection criteria.

4.6.1 Overview of existing techniques and placement of SIC

Various model selection criteria have been proposed so far. Here, we categorize them into six groups depending on the type of the error measure (Figure 4.3): 1. generalization error based criteria for average evaluation, 2. generalization error based criteria for worst evaluation, 3. predictive training error based criteria, 4. Bayesian statistics based criteria, 5. stochastic complexity based criteria, and 6. heuristics induced criteria.

Since the generalization measure J_G adopted by SIC is averaged over the noise (see Eq.(4.2)), SIC is a generalization error based criterion for average evaluation.

Based on the above categorization, we shall review the existing model selection methods, and investigate the relation to SIC in detail.

1. **Generalization error based criteria for average evaluation**
   - Akaike's information criterion (AIC) (Akaike, 1974)
   - AIC for unrealizable learning target function (Takeuchi, 1976)
   - AIC for general loss (Murata *et al.*, 1994; Konishi & Kitagawa, 1996)
   - Finite correction of AIC (Sugiura, 1978)
   - Bootstrap correction of AIC (Ishiguro *et al.*, 1997; Shibata, 1997)
   - AIC for predictive inference (Satoh, 1997; Shimodaira, 1997)
   - Subspace information criterion (SIC)
2. **Generalization error based criteria for worst evaluation**
   - Structural risk minimization principle (Vapnik, 1995; Cherkassky *et al.*, 1999)
3. **Predictive training error based criteria**
   - C_P (Mallows, 1964), C_L (Mallows, 1973)
   - Cross-validation (Mosteller & Wallace, 1963; Allen, 1974)
   - Generalized cross-validation (Craven & Wahba, 1979)
4. **Bayesian statistics based criteria**
   - Bayesian information criterion (Schwarz, 1978)
   - A Bayesian information criterion (Akaike, 1980)
   - Evidence framework (MacKay, 1992)
5. **Stochastic complexity based criteria**
   - Minimum description length criterion (Rissanen, 1978)
   - Extended stochastic complexity (Yamanishi, 1998)
6. **Heuristics induced criteria**
   - Discrepancy principle (Groetsch, 1984; Morozov, 1993)
   - Metric based criterion (Schuurmans, 1997)

Figure 4.3: Categorization of model selection criteria.

4.6.2 Generalization error based criteria for average evaluation

Akaike's information criterion (AIC) (Akaike [1]) is one of the most eminent methods of the generalization error based criteria for average evaluation. Many successful applications of AIC to real world problems have been reported (e.g. Bozdogan [19], Akaike & Kitagawa [3][4], Kitagawa & Gersch [62]). AIC is a model selection criterion for maximum likelihood estimation, which is equivalent to LMS learning in the case of Gaussian noise. AIC for linear regression models with Gaussian noise (i.e., the learning result function $\hat{f}_S$ is given by Eqs.(4.36) and (4.37)) is expressed as (see Sakamoto *et al.* [106])

$$\mathrm{AIC}[S] = M \log J_{TE}^{S} + 2(\dim S + 1), \tag{4.81}$$

where J_{TE}^{S} is the training error of $\hat{f}_S(\boldsymbol{x})$ defined as

$$J_{TE}^{S} = \frac{1}{M} \sum_{m=1}^{M} \left| \hat{f}_S(\boldsymbol{x}_m) - y_m \right|^2. \tag{4.82}$$

AIC is an estimate of an essential part of the expected Kullback-Leibler information (Kullback & Leibler [65]), which is also referred to as the expected log-likelihood. This measure is conceptually equivalent to (see Murata *et al.* [82])

$$\mathrm{E}_{\{x_m\}} \mathrm{E}_{\boldsymbol{\epsilon}} \int \left| \hat{f}_\theta(\boldsymbol{u}) - f(\boldsymbol{u}) \right|^2 p(\boldsymbol{u}) d\boldsymbol{u}, \tag{4.83}$$

where $p(\cdot)$ is the (possibly unknown) probability density function of training and future sample points. $\mathrm{E}_{\{x_m\}}$ denotes the ensemble average over all possible training sample points $\{\boldsymbol{x}_m\}_{m=1}^{M}$. The integral with respect to $\boldsymbol{u}$ corresponds to the expectation over future sample points. In contrast, the generalization measure J_G adopted by SIC is typically expressed as

$$\mathrm{E}_{\boldsymbol{\epsilon}} \int \left| \hat{f}_\theta(\boldsymbol{u}) - f(\boldsymbol{u}) \right|^2 p(\boldsymbol{u}) d\boldsymbol{u}, \tag{4.84}$$

where $p(\cdot)$ is the probability density function of future sample points $\boldsymbol{u}$, which is generally different from the probability density function of training sample points $\{\boldsymbol{x}_m\}_{m=1}^{M}$.

The ideal generalization measure may be

$$\int \left| \hat{f}_\theta(\boldsymbol{u}) - f(\boldsymbol{u}) \right|^2 p(\boldsymbol{u}) d\boldsymbol{u}. \tag{4.85}$$

Compared with Eq.(4.85), the generalization measure J_G adopted by SIC is equivalent to the expectation of Eq.(4.85) over the noise, where the covariance operator of $p(\cdot)$ is

assumed to be known (see Eq.(4.22)). In contrast, the generalization measure adopted by AIC further takes the expectation over training sample points $\{\boldsymbol{x}_m\}_{m=1}^M$ under assumptions that training sample points $\{\boldsymbol{x}_m\}_{m=1}^M$ and future sample points $\boldsymbol{u}$ are assumed to be independently drawn from the same, possibly unknown distribution. Then the averaged terms over training sample points are replaced with particular values calculated by one given training set (see Murata *et al* [82]). The fact that the replacement of the averaged terms is unnecessary for SIC is expected to result in a more accurate estimate of the generalization error than AIC.

The restriction of AIC that training sample points $\{\boldsymbol{x}_m\}_{m=1}^M$ and future sample points $\boldsymbol{u}$ are independently drawn from the same distribution is removed by Shimodaira [119] and Satoh [108][109] (see also Barron [12]). However, the distributions of both future and training sample points should be known instead. In contrast, SIC does not require the information of the distribution of training sample points.

AIC assumes that the model S includes the learning target function $f(\boldsymbol{x})$ while SIC assumes only $f \in H$, i.e., S does not necessarily contain $f(\boldsymbol{x})$. Takeuchi [133] extended AIC to be applicable to models which do not include $f(\boldsymbol{x})$. This criterion is called Takeuchi's modification of AIC (TIC) (see also Stone [126], Shibata [117]).

The learning method with which AIC and TIC can deal is restricted to the maximum likelihood estimation. In contrast, SIC can consistently treat various learning methods including LMS learning, regularization learning, projection learning (Ogawa [88]), and parametric projection learning (Oja & Ogawa [95]). Murata *et al.* [82] relaxed the restriction of the maximum likelihood estimation and proposed the network information criterion (NIC), which can deal with any differentiable loss functions such as the log-loss or squared loss. As shown in Section 4.7, NIC with the squared loss for linear regression models (i.e., the learning result function $\hat{f}_S$ is given by Eqs.(4.36) and (4.37)) is equivalent to C_P (Mallows [72][73]) (see Section 4.6.4 for the definition of C_P). In NIC, a regularization term can also be included in the loss function. NIC with the squared loss and a quadratic regularizer for linear regression models (i.e., the learning result function $\hat{f}_{T,\alpha}$ is given by Eqs.(4.53) and (4.54)) is expressed as (Murata [81])

$$\mathrm{NIC}[T,\alpha] = J_{TE}^{T,\alpha} + \frac{1}{2M}\mathrm{tr}GF^{-1}, \tag{4.86}$$

where

$$J_{TE}^{T,\alpha} = \frac{1}{M}\sum_{m=1}^{M}\left|\hat{f}_{T,\alpha}(\boldsymbol{x}_m) - y_m\right|^2, \tag{4.87}$$

$$[G]_{pp'} = \frac{1}{M}\sum_{m=1}^{M}\left(\partial_p\left(\left|\hat{f}_{T,\alpha}(\boldsymbol{x}_m) - y_m\right|^2 + \frac{\alpha}{M}\|T\hat{f}_{T,\alpha}\|^2\right)\right.$$
$$\left.\times\,\partial_{p'}\left(\left|\hat{f}_{T,\alpha}(\boldsymbol{x}_m) - y_m\right|^2 + \frac{\alpha}{M}\|T\hat{f}_{T,\alpha}\|^2\right)\right), \tag{4.88}$$

$$[F]_{pp'} = \frac{1}{M}\sum_{m=1}^{M}\partial_p\partial_{p'}\left(\left|\hat{f}_{T,\alpha}(\boldsymbol{x}_m) - y_m\right|^2 + \frac{\alpha}{M}\|T\hat{f}_{T,\alpha}\|^2\right). \tag{4.89}$$

Here, ∂_p denotes the partial derivative operator with respect to the p-th coefficient w_p of the learning result function $\hat{f}_{T,\alpha}(\boldsymbol{x})$ expressed as

$$\hat{f}_{T,\alpha}(\boldsymbol{x}) = \sum_{p=1}^{\mu} w_p\varphi_p(\boldsymbol{x}), \tag{4.90}$$

where $\{\varphi_p(\boldsymbol{x})\}_{p=1}^{\mu}$ are basis functions in H. Konishi and Kitagawa [64] derived a generalization of TIC called the generalized information criterion (GIC). GIC can deal with a class of learning methods represented by statistical functionals under the Kullback-Leibler information.

AIC and above mentioned derivatives give an asymptotic unbiased estimate of the generalization error. This implies that when the number of training examples is small, those criteria are no longer valid. In contrast, SIC gives an exact unbiased estimate of the generalization error with a finite number of training examples (see Lemma 4.3). Therefore, SIC is expected to work well even when the number of training examples is small.

Two approaches have been taken for overcoming the weakness of AIC, i.e., it requires a large number of training examples. One is to calculate an exact unbiased estimate of the expected log-likelihood. Sugiura [127] proposed the corrected AIC (cAIC) for linear regression models with maximum likelihood estimation. cAIC for Gaussian noise (i.e., the learning result function $\hat{f}_S$ is given by Eqs.(4.36) and (4.37)) is expressed as

$$\mathrm{cAIC}[S] = M\log J_{TE}^{S} + \frac{2(\dim S + 1)M}{M - \dim S - 2}, \tag{4.91}$$

where J_{TE}^{S} is the training error of $\hat{f}_S(\boldsymbol{x})$ defined by Eq.(4.82). Arising from Sugiura's correction, a lot of research in this line were conducted (e.g. Hurvich & Tsai [52][53][54], Noda *et al.* [85], Fujikoshi & Satoh [36], Satoh *et al.* [107], Hurvich *et al.* [51], Simonoff

[121], McQuarrie & Tsai [76]). The other approach to overcoming the weakness of AIC is to use the bootstrap method (Efron [31], Efron & Tibshirani [33]) for numerically evaluating the bias when the expected log-likelihood is estimated by the log-likelihood. The idea of the bootstrap bias correction is first introduced by Wong [148] and Efron [32], and then it is formalized as a model selection criterion by Ishiguro *et al.* [55] (see also Davison & Hinkley [30], Cavanaugh & Shumway [21], Shibata [118]).

In the derivation of AIC, terms which are not dominant for model selection are neglected (see e.g. Murata *et al.* [82]). Due to the fact, AIC is effective only in the selection of nested models (see Takeuchi [134], Murata *et al.* [82]):

$$S_1 \subset S_2 \subset \cdots. \tag{4.92}$$

SIC is restricted to the case where the learning result function $\hat{f}_\theta$ is given by using a linear operators X_θ as Eq.(4.5). In contrast, AIC is applicable to non-linear operators. However, Hagiwara *et al.* [47] showed that the conditions assumed in the derivation of AIC do not hold if the model has singular points, i.e., the Fisher information matrix is degenerated (see also Fukumizu [37]). This mean that AIC can not be applied to the problem of determining the number of hidden units in hierarchical models such as neural networks (see also Section 7.2.8).

AIC assumes that training examples are independently and identically distributed (*i.i.d.*). In contrast, SIC can deal with the correlated noise if the noise covariance matrix Q is available.

4.6.3 Generalization error based criteria for worst evaluation

The structural risk minimization principle (Vapnik [140][141]) evaluates a probabilistic upper bound of the risk functional defined as

$$\int \left| \hat{f}_\theta(\boldsymbol{u}) - f(\boldsymbol{u}) \right|^2 p(\boldsymbol{u}) d\boldsymbol{u}, \tag{4.93}$$

where $p(\cdot)$ is the (possibly unknown) probability density function of training and future sample points. One of the probabilistic upper bounds is given as

$$J^\theta_{TE} \Big/ \max\left(0, 1 - c\sqrt{\frac{h(\ln(\frac{aM}{h}+1) - \ln\eta}{M}}\right), \tag{4.94}$$

where a and c are some constants, h is the VC-dimension (see Vapnik [139][140][141]) of the model θ, and J_{TE}^{θ} is the training error of $\hat{f}_{\theta}(x)$ defined as

$$J_{TE}^{\theta} = \frac{1}{M}\sum_{m=1}^{M}\left|\hat{f}_{\theta}(\boldsymbol{x}_m) - y_m\right|^2. \tag{4.95}$$

Eq.(4.94) is a probabilistic upper bound of Eq.(4.93) with probability $1-\eta$.

Cherkassky *et al.* [23] heuristically let $a = 1$, $c = 1$, and $\eta = \frac{1}{\sqrt{M}}$ in Eq.(4.94), and named the bound Vapnik's measure (VM):

$$\mathrm{VM}[\theta] = J_{TE}^{\theta} \Big/ \max\left(0, 1 - \sqrt{p - p\log p + \frac{\log M}{2M}}\right), \tag{4.96}$$

where

$$p = \frac{h}{M}. \tag{4.97}$$

For calculating VM, the VC-dimension h of the model θ should be explicitly calculated. In the case of LMS learning for linear regression models (i.e., the learning result function $\hat{f}_S$ is given by Eqs.(4.36) and (4.37)), the VC-dimension of the subspace model S is given as

$$h = \dim S. \tag{4.98}$$

However, evaluating the VC-dimension is not generally easy, and it is often heuristically determined. For example, in the case of regularization learning with a quadratic regularizer for linear regression models (i.e., the learning result function $\hat{f}_{T,\alpha}$ is given by Eqs.(4.53) and (4.54)), the VC-dimension is heuristically put as (Cherkassky *et al.* [23], see also Shao *et al.* [116])

$$h = \mathrm{tr}B^{*}BD_{T,\alpha}^{-1}, \tag{4.99}$$

where B and $D_{T,\alpha}$ are given by Eqs.(4.40) and (4.56), respectively.

4.6.4 Predictive training error based criteria

The *predictive training error* J_{PTE} is often used as the error measure (Figure 4.4):

$$J_{PTE} = \mathrm{E}_{\boldsymbol{\epsilon}}\frac{1}{M}\sum_{m=1}^{M}\left|\hat{f}_{\theta}(\boldsymbol{x}_m) - f(\boldsymbol{x}_m)\right|^2. \tag{4.100}$$

Note that J_{PTE} can be obtained by substituting the empirical distribution of training sample points $\{\boldsymbol{x}_m\}_{m=1}^{M}$ into $w(\boldsymbol{u})$ in Eq.(4.84), i.e.,

$$w(\boldsymbol{u}) = \begin{cases} \frac{1}{M} & \text{if } \boldsymbol{u} = \boldsymbol{x}_m, \\ 0 & \text{otherwise.} \end{cases} \tag{4.101}$$

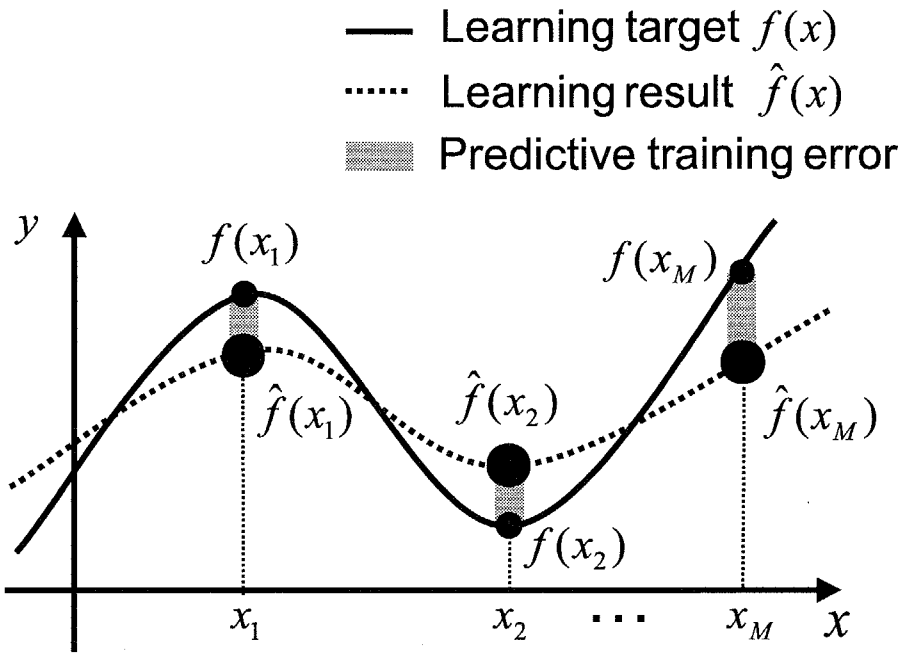


Figure 4.4: Predictive training error.

Mallows [72] proposed C_P for LMS learning with linear regression models (i.e., the learning result function $\hat{f}_S$ is given by Eqs.(4.36) and (4.37)):

$$C_P[S] = J_{TE}^S + \frac{2\hat{\sigma}^2 \dim S}{M} - \hat{\sigma}^2, \tag{4.102}$$

where J_{TE}^S is the training error of $\hat{f}_S$ given by Eq.(4.82), $\hat{\sigma}^2$ is an estimate of the noise variance σ^2, and S is a LMS learning model, i.e., a reproducing kernel Hilbert space.

Mallows [73] extended the range of application of C_P to the selection of arbitrary models expressed by using a linear operator X_θ as Eq.(4.5). It is called C_L or the unbiased risk estimate (Wahba [145]):

$$C_L[\theta] = J_{TE}^\theta - \frac{\hat{\sigma}^2}{M}\mathrm{tr}(AX_\theta - I_M)(AX_\theta - I_M)^* + \frac{\hat{\sigma}^2}{M}\mathrm{tr}AX_\theta(AX_\theta)^*, \tag{4.103}$$

where J_{TE}^θ is the training error of $\hat{f}_\theta$ given by Eq.(4.95) and A is given by Eq.(4.25). C_L for regularization learning with a quadratic regularizer for linear regression models (i.e., the learning result function $\hat{f}_{T,\alpha}$ is given by Eqs.(4.53) and (4.54)) is expressed as

$$C_L[T,\alpha] = J_{TE}^{T,\alpha} + \frac{2\hat{\sigma}^2}{M}\mathrm{tr}B^*BD_{T,\alpha}^{-1} - \hat{\sigma}^2, \tag{4.104}$$

where $J_{TE}^{T,\alpha}$, B, and $D_{T,\alpha}$ are given by Eqs.(4.87), (4.40), and (4.56), respectively. Note

that C_P and C_L are unbiased estimates of the predictive training error J_{PTE} if the noise covariance matrix Q is given by $Q = \sigma^2 I_M$ with $\sigma^2 > 0$ and $\hat{\sigma}^2$ is obtained by Eq.(4.31).

The approximation scheme used in SIC is similar to C_P and C_L but the error measure is different: SIC is an estimate of the generalization error J_G while C_P and C_L are estimates of the predictive training error J_{PTE}. In C_P and C_L, the sample value vector $\boldsymbol{y}$ is used as an unbiased estimate of the ideal sample value vector $\boldsymbol{z}$ given by Eq.(4.12). In contrast, SIC assumes the availability of an unbiased estimate $\hat{f}_u(\boldsymbol{x})$ of the learning target function $f(\boldsymbol{x})$. $\hat{f}_u(\boldsymbol{x})$ plays a similar role to $\boldsymbol{y}$ in C_P and C_L.

Leave-one-out cross-validation (CV) (e.g. Mosteller & Wallace [78], Allen [6], Orr [96], Vapnik & Chapelle [138]) adopts the leave-one-out error as the error measure:

$$\mathrm{CV}[\theta] = \frac{1}{M}\sum_{m=1}^{M}\left|\hat{f}_\theta^{(m)}(\boldsymbol{x}_m) - y_m\right|^2, \tag{4.105}$$

where $\hat{f}_\theta^{(m)}$ denotes the learning result function obtained from the training examples without $(\boldsymbol{x}_m, y_m)$.

In the case of regularization learning with a quadratic regularizer for linear regression models (i.e., the learning result function $\hat{f}_{T,\alpha}$ is given by Eqs.(4.53) and (4.54)), a closed form expression of Eq.(4.105) is expressed as (see Orr [96])

$$\mathrm{CV}[T,\alpha] = \frac{1}{M}\|(\mathrm{diag}(Z_{T,\alpha}))^{-1} Z_{T,\alpha}\boldsymbol{y}\|^2, \tag{4.106}$$

Here, $Z_{T,\alpha}$ is an M-dimensional matrix defined as

$$Z_{T,\alpha} = I_M - B D_{T,\alpha}^{-1} B^*, \tag{4.107}$$

where B and $D_{T,\alpha}$ are given by Eqs.(4.40) and (4.56), respectively. The matrix '$\mathrm{diag}(Z_{T,\alpha})$' is the same size and has the same diagonal as $Z_{T,\alpha}$ but is zero off the diagonal.

Similarly, in the case of LMS learning for linear regression models (i.e., the learning result function $\hat{f}_S$ is given by Eqs.(4.36) and (4.37)), a closed form expression of Eq.(4.105) for a subspace model S is given as

$$\mathrm{CV}[S] = \frac{1}{M}\|(\mathrm{diag}(W_S))^{-1} W_S\boldsymbol{y}\|^2, \tag{4.108}$$

where W_S is an M-dimensional matrix defined as

$$W_S = I_M - B_S B_S^{\dagger}. \tag{4.109}$$

B_S is given by Eq.(4.41).

Except for the above cases, the leave-one-out error defined by Eq.(4.105) should be generally calculated by executing the learning procedure M times. This requires a lot of computation time when M is large.

Stone [125] showed that the model selection by CV is asymptotically equivalent to that by AIC (see also Amari *et al.* [10] for asymptotic analysis). Although it is known that CV practically works well, its mechanism in small sample cases is not well recognized yet.

Craven and Wahba [28] proposed the generalized cross-validation (GCV) that can be calculated easier than CV. GCV is applicable to regularization learning with a quadratic regularizer for linear regression models (i.e., the learning result function $\hat{f}_{T,\alpha}$ is given by Eqs.(4.53) and (4.54)), and it is given as

$$\mathrm{GCV}[T,\alpha] = \frac{J_{TE}^{T,\alpha}}{\left(1-\frac{1}{M}\mathrm{tr}B^*BD_{T,\alpha}^{-1}\right)^2}, \tag{4.110}$$

where $J_{TE}^{T,\alpha}$ is the training error of $\hat{f}_{T,\alpha}$ defined by Eq.(4.87), B is given by Eq.(4.40), and $D_{T,\alpha}$ is given by Eq.(4.56). GCV is a simplified version of CV with '$\mathrm{diag}(Z_{T,\alpha})$' in Eq.(4.106) replaced by '$(\frac{1}{M}\mathrm{tr}Z_{T,\alpha})I_M$' (see Orr [96]). Li [67] showed the asymptotic optimality of C_L and GCV, i.e., they asymptotically select the model that minimizes the predictive training error J_{PTE} defined by Eq.(4.100) (see also Wahba [145]).

The effectiveness of the predictive training error based methods relies on the expectation that reducing the predictive training error is equivalent to reducing the generalization error (see e.g. Mallows [73]). However, it is not generally true. This is the limitation of the predictive training error based methods.

4.6.5 Bayesian statistics based criteria

The probability of obtaining the learning result function $\hat{f}$ given training examples $\{(\boldsymbol{x}_m, y_m)\}_{m=1}^M$ and a model θ is expressed by using the Bayes rule as

$$P(\hat{f}|\{(\boldsymbol{x}_m, y_m)\}_{m=1}^M, \theta) = \frac{P(\{(\boldsymbol{x}_m, y_m)\}_{m=1}^M|\hat{f},\theta)P(\hat{f}|\theta)}{P(\{(\boldsymbol{x}_m, y_m)\}_{m=1}^M|\theta)}. \tag{4.111}$$

In words [68]:

$$\text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Evidence}}. \tag{4.112}$$

The Bayesian statistics based criteria select the model θ that maximizes the evidence. Note that the evidence is expressed as

$$P(\{(\boldsymbol{x}_m, y_m)\}_{m=1}^M|\theta) = \int P(\{(\boldsymbol{x}_m, y_m)\}_{m=1}^M|\hat{f}, \theta)P(\hat{f}|\theta)d\hat{f}. \tag{4.113}$$

Using the asymptotic approximation, Schwarz [115] gave an estimate of $-2\log P(\{(\boldsymbol{x}_m, y_m)\}_{m=1}^M|\theta)$ (the log-evidence multiplied by -2) for selecting maximum likelihood models. This estimate is called the the Bayesian information criterion (BIC). BIC for linear regression models with Gaussian noise (i.e., the learning result function $\hat{f}_S$ is given by Eqs.(4.36) and (4.37)) is expressed as

$$\mathrm{BIC}[S] = M\log J_{TE}^S + (\dim S + 1)\log M, \tag{4.114}$$

where J_{TE}^S is the training error of $\hat{f}_S(\boldsymbol{x})$ defined by Eq.(4.82). Note that the minimum description length (MDL) criterion (Rissanen [99][100][101]), which is derived in the light of information theory, is also the same form as BIC. The advantage of BIC is that it does not depend on the prior distribution of the learning result function. Therefore, it can be calculated without the knowledge of the prior distribution. However, due to the asymptotic approximation, BIC is valid only when a large number of training examples is available.

It is known that regularization learning can be regarded as the Bayesian inference (see e.g. Akaike [2]). In the case of Gaussian noise with the noise covariance matrix $Q = \sigma^2 I_M$ ($\sigma^2 > 0$), the likelihood is proportional to

$$\exp\left(-\frac{1}{2\sigma^2}\sum_{m=1}^M \left|\hat{f}(\boldsymbol{x}_m) - y_m\right|^2\right). \tag{4.115}$$

If the prior distribution of the learning result function $\hat{f}$ is proportional to

$$\exp\left(-\frac{1}{2\sigma^2}\alpha\|T\hat{f}\|^2\right), \tag{4.116}$$

then the log-posterior is expressed by using some constant c as

$$\log P(\hat{f}|\{(\boldsymbol{x}_m, y_m)\}_{m=1}^M, \theta) = \left(\sum_{m=1}^M \left|\hat{f}(\boldsymbol{x}_m) - y_m\right|^2\right) + \alpha\|T\hat{f}\|^2 + c \tag{4.117}$$

since the evidence is a constant for fixed $\{(\boldsymbol{x}_m, y_m)\}_{m=1}^M$ and θ. This means that maximizing the log-posterior with respect to the learning result function is equivalent to

minimizing the regularization learning criterion J_R defined by (4.52). Note that in the above interpretation, the regularization operator T and regularization parameter α control the prior distribution of the learning result function. In the Bayesian terminology, they are called the *hyper parameters.*

Akaike [2] named $-2\log P(\{(\boldsymbol{x}_m, y_m)\}_{m=1}^M|\theta)$ (the log-evidence multiplied by -2) a Bayesian information criterion (ABIC), and proposed using it for selecting the regularization operator and regularization parameter. A practical expression of ABIC for regularization learning with quadratic regularizers for linear regression models (i.e., the learning result function $\hat{f}_{T,\alpha}$ is given by Eqs.(4.53) and (4.54)) is given as

$$\mathrm{ABIC}[T, \alpha] = M \log \frac{J_R^{T,\alpha}}{M} + \log\det(D_{T,\alpha}) - \log\det(\alpha W^* T^* T W), \tag{4.118}$$

where $J_R^{T,\alpha}$ is defined as

$$J_R^{T,\alpha} = \sum_{m=1}^{M} \left| \hat{f}_{T,\alpha}(\boldsymbol{x}_m) - y_m \right|^2 + \alpha \|T \hat{f}_{T,\alpha}\|^2. \tag{4.119}$$

'det(·)' denotes the determinant of a matrix, and $D_{T,\alpha}$ and W are defined by Eqs.(4.56) and (4.42), respectively. When T is fixed and only α is optimized, ABIC is reduced to

$$\mathrm{ABIC}[\alpha] = M \log \frac{J_R^{T,\alpha}}{M} + \log\det(D_{T,\alpha}) - \dim H \log \alpha. \tag{4.120}$$

MacKay [68][69][71] also proposed a similar method to ABIC. Although it is reported that Bayesian statistics based criteria experimentally work well, their effectiveness is not theoretically sure since they do not directly evaluate the generalization error itself.

4.6.6 Heuristics induced criteria

One of the classic approaches to determining the regularization parameter is based on the discrepancy principle (Groetsch [45], Morozov [77], Kunisch & Zou [66]). The discrepancy principle asserts that the training error should be equal to the noise variance. A heuristic motivation for this principle is that it does not make sense to ask for an estimation with the training error less than the noise variance since only the noisy sample values are available (Groetsch [45]). However, the relation between the discrepancy principle and generalization capability is not sure.

Schuurmans [114] proposed a metric based criterion. Although its effectiveness is experimentally evaluated, its theoretical analysis is not still enough.

4.7 Approximated SIC

In this section, we derive an approximation of SIC named the *approximated SIC* (ASIC). We assume through this section that X_u in SIC is obtained by Eq.(4.30).

4.7.1 Derivation of approximated SIC

When $\frac{1}{M}A^*A = I_H$ where I_H denotes the identity operator on H, SIC defined by Eq.(4.18) is reduced to as follows.

$$\begin{aligned}
\mathrm{SIC}[\theta] &= \|(X_\theta - A^\dagger)\boldsymbol{y}\|^2 - \mathrm{tr}(X_\theta - A^\dagger)Q(X_\theta - A^\dagger)^* + \mathrm{tr}X_\theta Q X_\theta^* \\
&= \frac{1}{M}\langle A^*A(X_\theta - A^\dagger)\boldsymbol{y}, (X_\theta - A^\dagger)\boldsymbol{y}\rangle \\
&\quad - \frac{1}{M}\mathrm{tr}A^*A(X_\theta - A^\dagger)Q(X_\theta - A^\dagger)^* + \frac{1}{M}\mathrm{tr}A^*AX_\theta Q X_\theta^* \\
&= \frac{1}{M}\|AX_\theta\boldsymbol{y} - AA^\dagger\boldsymbol{y}\|^2 - \frac{1}{M}\mathrm{tr}(AX_\theta - AA^\dagger)Q(AX_\theta - AA^\dagger)^* \\
&\quad + \frac{1}{M}\mathrm{tr}AX_\theta Q(AX_\theta)^*. \qquad (4.121)
\end{aligned}$$

It follows from Eqs.(4.5) and (4.8) that the first term in Eq.(4.121) yields

$$\begin{aligned}
\frac{1}{M}\|AX_\theta\boldsymbol{y} - AA^\dagger\boldsymbol{y}\|^2 &= \frac{1}{M}\|AX_\theta\boldsymbol{y}\|^2 - \frac{2}{M}\mathrm{Re}\langle AX_\theta\boldsymbol{y}, AA^\dagger\boldsymbol{y}\rangle + \frac{1}{M}\|AA^\dagger\boldsymbol{y}\|^2 \\
&= \frac{1}{M}\|AX_\theta\boldsymbol{y}\|^2 - \frac{2}{M}\mathrm{Re}\langle AX_\theta\boldsymbol{y}, \boldsymbol{y}\rangle + \frac{1}{M}\|\boldsymbol{y}\|^2 \\
&\quad - \frac{1}{M}\|\boldsymbol{y}\|^2 + \frac{1}{M}\|AA^\dagger\boldsymbol{y}\|^2 \\
&= \frac{1}{M}\|AX_\theta\boldsymbol{y} - \boldsymbol{y}\|^2 - \frac{1}{M}\|AA^\dagger\boldsymbol{y} - \boldsymbol{y}\|^2 \\
&= \frac{1}{M}\|A\hat{f}_\theta - \boldsymbol{y}\|^2 - \frac{1}{M}\|A\hat{f}_u - \boldsymbol{y}\|^2 \\
&= J_{TE}^\theta - J_{TE}^u, \qquad (4.122)
\end{aligned}$$

where J_{TE}^θ is the training error of $\hat{f}_\theta$ defined by Eq.(4.95), and J_{TE}^u is the training error of $\hat{f}_u$ defined as

$$J_{TE}^u = \frac{1}{M}\sum_{m=1}^{M}\left|\hat{f}_u(\boldsymbol{x}_m) - y_m\right|^2. \qquad (4.123)$$

The second term in Eq.(4.121) yields

$$\begin{aligned}
&-\frac{1}{M}\mathrm{tr}(AX_\theta - AA^\dagger)Q(AX_\theta - AA^\dagger)^* \\
&\quad = -\frac{1}{M}\mathrm{tr}\left(AX_\theta Q(AX_\theta)^* - AX_\theta QAA^\dagger\right.
\end{aligned}$$

$$
\begin{aligned}
&-AA^{\dagger}Q(AX_{\theta})^{*}+AA^{\dagger}QAA^{\dagger}) \\
&=-\frac{1}{M}\mathrm{tr}\left(AX_{\theta}Q(AX_{\theta})^{*}-QAX_{\theta}-(AX_{\theta})^{*}Q+QAA^{\dagger}\right). \qquad (4.124)
\end{aligned}
$$

If J_{TE}^{u} in Eq.(4.122) and $-\frac{1}{M}\mathrm{tr}QAA^{\dagger}$ in Eq.(4.124) are ignored since they are irrelevant to model selection, then we have the following criterion.

Definition 4.16 (Approximated SIC) *The following functional* ASIC$[\theta]$ *is called the approximated subspace information criterion (ASIC) for a model* θ*:*

$$
\mathrm{ASIC}[\theta]=J_{TE}^{\theta}+\frac{2}{M}\mathrm{Re}\ \mathrm{tr}QAX_{\theta}. \qquad (4.125)
$$

ASIC can be used as an approximation of SIC if $\frac{1}{M}A^{*}A\approx I_{H}$. Note that $\frac{1}{M}A^{*}A=I_{H}$ strictly holds if H is a trigonometric polynomial space and $\{\boldsymbol{x}_{m}\}_{m=1}^{M}$ are fixed to regular intervals in the domain (see Section 5.3.4 for detail). The advantage of using ASIC is that its calculation is easier than the original SIC. Indeed, when the noise covariance matrix Q is estimated by Eq.(4.31) and LMS learning is adopted, ASIC yields

$$
J_{TE}^{\theta}+\frac{2\hat{\sigma}^{2}}{M}\dim S. \qquad (4.126)
$$

This can be verified from

$$
AX_{\theta}=AA_{S}^{\dagger}=AA_{S}^{*}(A_{S}A_{S}^{*})^{\dagger}=A_{S}A_{S}^{*}(A_{S}A_{S}^{*})^{\dagger}=P_{\mathcal{R}(A_{S})}, \qquad (4.127)
$$

where $P_{\mathcal{R}(A_{S})}$ denotes the orthogonal projection operator onto the range of A_{S}.

4.7.2 Relation to C_L

C_L (Mallows [73]) given by Eq.(4.103) is a model selection criterion for selecting the model that minimizes the predictive training error (see Section 4.6.4). If Q in Eq.(4.125) is estimated by Eq.(4.31), then it holds that

$$
\mathrm{ASIC}[\theta]=C_{L}[\theta]-\hat{\sigma}^{2}. \qquad (4.128)
$$

This implies that C_L essentially agrees with ASIC since the second term in Eq.(4.128) is irrelevant to model selection. Therefore, C_L can be regarded as an approximation of SIC if $\frac{1}{M}A^{*}A\approx I_{H}$. Conversely, it shows that, if $\frac{1}{M}A^{*}A\approx I_{H}$, C_L works well despite the fact that C_L does not directly take the generalization error J_G into account.

4.7.3 Relation to network information criterion

Akaike's information criterion (AIC) (Akaike [1]) is a model selection criterion for selecting the model that minimizes the Kullback-Leibler information (Kullback & Leibler [65]). The network information criterion (NIC) (Murata *et al.* [82]) is a generalized AIC in which any differentiable loss function can be handled (see Section 4.6.2). Let us assume that the sample value y_m and noise ϵ_m are real since NIC is derived under this condition. In this case, 'Re' in Eq.(4.125) is not required. NIC with the squared loss for linear regression models (i.e., the learning result function $\hat{f}_S$ is given by Eqs.(4.36) and (4.37)) is expressed as

$$\mathrm{NIC}[S] = J_{TE}^S + \frac{2}{M}\mathrm{tr}\hat{Q}_S A_S A_S^\dagger, \tag{4.129}$$

where J_{TE}^S is the training error of $\hat{f}_S(\boldsymbol{x})$ defined by Eq.(4.82) and $\hat{Q}_S$ is an M-dimensional diagonal matrix with the m-th diagonal element being $(y_m - \hat{f}_S(\boldsymbol{x}_m))^2$:

$$\hat{Q}_S = \mathrm{diag}\left((y_1 - \hat{f}_S(\boldsymbol{x}_1))^2, (y_2 - \hat{f}_S(\boldsymbol{x}_2))^2, \ldots, (y_M - \hat{f}_S(\boldsymbol{x}_M))^2\right). \tag{4.130}$$

On the other hand, ASIC for LMS learning is expressed as

$$J_{TE}^S + \frac{2}{M}\mathrm{tr}QA_S A_S^\dagger \tag{4.131}$$

because of Eq.(4.127). Eqs.(4.129) and (4.131) imply that NIC with the squared loss is essentially equivalent to ASIC with LMS learning except that the noise covariance matrix Q is estimated by $\hat{Q}_S$. Therefore, NIC with the squared loss can be used as an approximation of SIC with LMS learning if $\frac{1}{M}A^*A \approx I_H$. Conversely, it shows that, if $\frac{1}{M}A^*A \approx I_H$, NIC with the squared loss works well with a small number of training examples despite the fact that NIC is derived by making use of asymptotic approximation.

4.7.4 Estimation methods of noise covariance matrix

Now we shall discuss the estimation methods of the noise covariance matrix Q. Let us assume that the noises $\{\epsilon_m\}_{m=1}^M$ are uncorrelated and let σ_m^2 be the noise variance of ϵ_m. In this case, Q is an M-dimensional diagonal matrix with the m-th diagonal element being σ_m^2:

$$Q = \mathrm{diag}\left(\sigma_1^2, \sigma_2^2, \ldots, \sigma_M^2\right). \tag{4.132}$$

In NIC, Q is estimated by the diagonal matrix $\hat{Q}_S$ (Eq.(4.130)) whether the subspace

model S is *faithful*[1] or not. If the model S is faithful, then $(y_m - \hat{f}_S(\boldsymbol{x}_m))^2$ tends to ϵ_m^2 since $\hat{f}_S(x)$ tends to the learning target function $f(\boldsymbol{x})$ as $M \to \infty$. Hence, $(y_m - \hat{f}_S(\boldsymbol{x}_m))^2$ can be naturally regarded as an estimate of the noise variance σ_m^2. However, when it comes to the unfaithful case, it is difficult to justify the validity of the estimate since $\hat{f}_S(\boldsymbol{x})$ does not converge to the learning target function $f(\boldsymbol{x})$. In contrast, estimation of the noise covariance matrix Q by Eq.(4.31) is not obtained with the model S but obtained with an unbiased learning result function $\hat{f}_u(\boldsymbol{x})$ based on the fact that the noise characteristic does not depend on models. Hence, if $\hat{f}_u(\boldsymbol{x})$ is available, then the noise variance can be reasonably estimated irrespective of the faithfulness of the subspace model S (see also Barron [12]). This idea can also be applied to NIC, i.e., $\hat{Q}_u$ obtained with a faithful model is commonly used as an estimate of the noise covariance matrix Q for all models:

$$\hat{Q}_u = \mathrm{diag}\left((y_1 - \hat{f}_u(\boldsymbol{x}_1))^2, (y_2 - \hat{f}_u(\boldsymbol{x}_2))^2, \ldots, (y_M - \hat{f}_u(\boldsymbol{x}_M))^2\right). \tag{4.133}$$

However, even with $\hat{Q}_u$, $(y_m - \hat{f}_u(\boldsymbol{x}_m))^2$ may not be a good estimate of σ_m^2 since it is estimated from only one training example $(\boldsymbol{x}_m, y_m)$. In contrast, Eq.(4.31) assumes that values $\{\sigma_m^2\}_{m=1}^M$ of the noise variance agree with each other:

$$\sigma_1^2 = \sigma_2^2 = \cdots = \sigma_M^2 = \sigma^2. \tag{4.134}$$

This implies that if values $\{\sigma_m^2\}_{m=1}^M$ of the noise variance are not so different, a good estimate of the noise covariance matrix Q may be obtained since the common noise variance σ^2 is estimated from M training examples $\{(\boldsymbol{x}_m, y_m)\}_{m=1}^M$.

4.8 Computer simulations

In this section, the effectiveness of SIC is experimentally demonstrated through computer simulations. The simulations are performed for LMS learning and regularization learning.

4.8.1 SIC for least mean squares learning

SIC for LMS learning given in Section 4.4 is experimentally evaluated.

[1] A model is said to be *faithful* if the learning target function can be expressed by the model (Murata *et al.* [82]).

4.8.1.1 Setting

Let the dimension L of the input vector $\boldsymbol{x}$ be 1 and S_n be a trigonometric polynomial space of order n (see Section 3.3.1), i.e., S_n is spanned by the functions

$$\left\{1, \sqrt{2}\sin px, \sqrt{2}\cos px\right\}_{p=1}^{n} \tag{4.135}$$

defined on $[-\pi, \pi]$, and the inner product is defined as

$$\langle f, g\rangle = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(x)\overline{g(x)}dx. \tag{4.136}$$

Note that the dimension of S_n is $2n+1$. We adopt S_{100} as H.

Let the learning target function $f(x)$ be

$$f(x) = \frac{1}{10}\sum_{p=1}^{50}(\sin px + \cos px). \tag{4.137}$$

Let the sample points $\{x_m\}_{m=1}^{M}$ be randomly created in the domain $[-\pi, \pi]$, and the noise ϵ_m be independently subject to the same normal distribution with mean 0 and variance σ^2:

$$\epsilon_m \sim N(0, \sigma^2). \tag{4.138}$$

In this case, the noise covariance matrix Q is given as

$$Q = \sigma^2 I_M. \tag{4.139}$$

The simulation is performed 100 times for $(M, \sigma^2) = (500, 0.2)$, $(250, 0.2)$, $(500, 0.6)$, and $(250, 0.6)$, with changing the noise $\{\epsilon_m\}_{m=1}^{M}$ in each trial.

We adopt LMS learning and let the set $\mathcal{M}$ of model candidates be

$$\mathcal{M} = \{S_0, S_{10}, S_{20}, \ldots, S_{100}\}. \tag{4.140}$$

We shall measure the error of a learning result function $\hat{f}_S(x)$ by

$$\mathrm{Error}[S] = \|\hat{f}_S - f\|^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi}\left|\hat{f}_S(x) - f(x)\right|^2 dx. \tag{4.141}$$

Let a function $\varphi_p(x)$ be

$$\varphi_p(x) = \begin{cases} 1 & \text{if } p = 1, \\ \sqrt{2}\sin(\frac{px}{2}) & \text{if } p \geq 2 \text{ and } p \text{ is even}, \\ \sqrt{2}\cos(\frac{(p-1)x}{2}) & \text{if } p \geq 3 \text{ and } p \text{ is odd}. \end{cases} \tag{4.142}$$

In this case, the covariance matrix U is reduced to the identity matrix. The Moore-Penrose generalized inverse is calculated by Eqs.(4.49) and (4.50) with $\gamma = 0.1$.

4.8.1.2 Comparison with existing model selection methods

The following model selection criteria are compared.

(a) Subspace information criterion (SIC): X_u is obtained by Eq.(4.30) and Q is estimated by Eq.(4.31). In this case, SIC for a model S is given by Eq.(4.45) with U being the identity matrix.

(b) C_P (Mallows [72][73]): C_P for a model S is given by Eq.(4.102).

(c) Leave-one-out cross-validation (CV): A closed form expression of the leave-one-out error for a model S is given by Eq.(4.108).

(d) Akaike's information criterion (AIC) (Akaike [1]): When the noise is subject to the normal distribution, AIC for a model S is expressed by Eq.(4.81).

(e) Corrected AIC (cAIC) (Sugiura [127]): When the noise is subject to the normal distribution, cAIC for a model S is expressed by Eq.(4.91).

(f) Bayesian information criterion (BIC) (Schwarz [115]): When the noise is subject to the normal distribution, BIC for a model S is expressed by Eq.(4.114). Note that the minimum description length (MDL) criterion (Rissanen [99][100][101]) is also given by Eq.(4.114).

(g) Vapnik's measure (VM) (Cherkassky *et al.* [23]): VM for a model S is given by Eq.(4.96) with h given by Eq.(4.98).

Figures 4.5, 4.6, 4.7, and 4.8 display the simulation results for $(M, \sigma^2) = (500, 0.2)$, $(250, 0.2)$, $(500, 0.6)$, and $(250, 0.6)$, respectively. The top eight graphs show the values of the error and model selection criteria corresponding to the order n of the model S_n (see Eq.(4.140)). The box plot notation specifies marks at 95, 75, 50, 25, and 5 percentiles of values. The solid line denotes the mean values. The bottom-left eight graphs show the distributions of the selected order n of models. 'OPT' indicates the optimal model that minimizes the error measured by Eq.(4.141). The bottom-right eight graphs show the distributions of the error obtained by the model selected by each criterion.

When $(M, \sigma^2) = (500, 0.2)$ (Figure 4.5), all model selection criteria work well.

When $(M, \sigma^2) = (250, 0.2)$ (Figure 4.6), SIC, CV, C_P, cAIC, and VM work well. In this case, AIC tends to select larger models and BIC (MDL) is inclined to select smaller

models, so they provide large errors. This may be caused since AIC and BIC (MDL) are derived under the assumption that the number M of training examples is very large.

When $(M, \sigma^2) = (500, 0.6)$ (Figure 4.7), SIC, CV, C_P, AIC, and cAIC work well. BIC (MDL) and VM show a tendency to select smaller models and they result in large errors. This implies that BIC (MDL) and VM are not robust against the noise.

Finally, when $(M, \sigma^2) = (250, 0.6)$ (Figure 4.8), only SIC can select reasonable models. In this case, C_P almost always selects S_{50}, AIC tends to select larger models, and other criteria tend to select smaller models. As a result, they give large errors. The results are summarized in Figure 4.9.

The simulation results show that SIC outperforms other model selection criteria especially when the number M of training examples is small and the noise variance σ^2 is large (see Figure 4.8). Although SIC almost always gives a very good estimate of the true error on average, its variance is rather large when $M = 250$ (Figures 4.6 and 4.8). However, the large variance of SIC may be dominated by terms that are irrelevant to model selection since SIC given by Eq.(4.45) is expressed as

$$\begin{aligned} \mathrm{SIC}[S] = \langle UB_S^\dagger \boldsymbol{y}, B_S^\dagger \boldsymbol{y} \rangle - 2\mathrm{Re}\, \langle UB_S^\dagger \boldsymbol{y}, B^\dagger \boldsymbol{y} \rangle + \langle UB^\dagger \boldsymbol{y}, B^\dagger \boldsymbol{y} \rangle \\ + 2\hat{\sigma}^2 \mathrm{Re}\, \mathrm{tr} UB_S^\dagger (B^\dagger)^* - \hat{\sigma}^2 \mathrm{tr} UB^\dagger (B^\dagger)^*, \end{aligned} \tag{4.143}$$

where the third and fifth terms are irrelevant to the model S.

It should be noted that C_P almost always selects the true model S_{50} in any cases. Here, the true model indicates the smallest model that includes the learning target function $f(x)$. This implies that C_P is more suitable for finding the true model than finding the model with the minimum generalization error.

4.8.1.3 Uniform noise

Let us investigate the robustness of SIC against non-Gaussian noise. We consider the same setting as Section 4.8.1.1 but the noise ϵ_m is subject to the uniform distribution on $[-0.3, 0.3]$.

The simulation results for $M = 500$ and 250 are displayed in Figures 4.10 and 4.11, respectively. The results show that SIC works well even in the uniform noise case.

4.8.1.4 Changing dimension of H

Now we investigate the influence of changing H when $(M, \sigma^2) = (250, 0.2)$. The simulation is performed with the same setting as Section 4.8.1.1 but H is changed as S_{120}, S_{100}, S_{80}, or S_{60}. Note that when H is S_{80} or S_{60}, we only consider the model candidates included in H (see Eq.(4.140)).

The simulation results displayed in Figure 4.12 show that the variance of SIC is reduced as H is small. This may be because $\hat{f}_u$ and $\hat{\sigma}^2$ tend to be accurate as H is small (see Eq.(4.143)). The performance of SIC is almost the same when H is S_{100}, S_{80}, or S_{60}. However, when $H = S_{120}$, the variance is rather large. This implies that when the dimension of H is very close to the number M of training examples (e.g. when $H = S_{120}$, $\dim H = 241$ and $M = 250$), SIC tends to be inaccurate.

4.8.1.5 Estimating U from unlabeled sample points

Let us investigate the robustness of SIC when the covariance matrix U (see Eq.(4.44)), which is assumed to be known, is estimated from unlabeled sample points $\{x'_m\}_{m=1}^{M'}$ (i.e., sample points without sample values $\{y'_m\}_{m=1}^{M'}$) as

$$[\hat{U}]_{p,p'} = \frac{1}{M'} \sum_{m=1}^{M'} \varphi_{p'}(x'_m)\overline{\varphi_p(x'_m)}. \tag{4.144}$$

Note that if the training sample points $\{x_m\}_{m=1}^{M}$ are used instead of unlabeled sample points, then SIC agrees with Mallows's C_P (see Section 4.6.4).

The simulation is performed with the same setting as Section 4.8.1.1 but U is estimated from M' unlabeled sample points subject to the uniform distribution on $[-\pi, \pi]$. M' is changed as 500, 250, 100, and 50. Figure 4.13 displays the simulation results when $(M, \sigma^2) = (500, 0.6)$. The simulation results show that the good performance of SIC is maintained as the number M' of unlabeled sample points is small. This implies that SIC will work well if only a rough estimate of U is available.

4.8.1.6 Unrealizable learning target function

Finally, we consider the case when the learning target function $f(x)$ is not included in H. The simulation is performed with the same setting as Section 4.8.1.1 but the learning target function $f(x)$ is the step function or $\frac{1}{1+x^2}$ defined on $[-\pi, \pi]$. Let the number M of training examples be 100. We adopt S_{20} as H. Let the set $\mathcal{M}$ of model candidates be

$\{S_0, S_2, S_4, \ldots, S_{20}\}$, which are included in H. We measure the error of a learning result function $\hat{f}_S(x)$ by using 1000 future sample points $\{u_j\}_{j=1}^{1000}$ randomly generated in $[-\pi, \pi]$ as

$$\text{Error}[S_n] = \frac{1}{1000} \sum_{j=1}^{1000} \left| \hat{f}_S(u_j) - f(u_j) \right|^2. \tag{4.145}$$

The simulation results with the learning target function being the step function for $(M, \sigma^2) = (100, 0.1)$ are shown in Figure 4.14. The simulation results with $f(x) = \frac{1}{1+x^2}$ for $(M, \sigma^2) = (100, 0.03)$ are shown in Figure 4.15. These results show that SIC seems still effective even in unrealizable cases as long as the Hilbert space H approximately includes the learning target function $f(x)$. However, further experiments may be needed to confirm the robustness against unrealizable learning target functions.

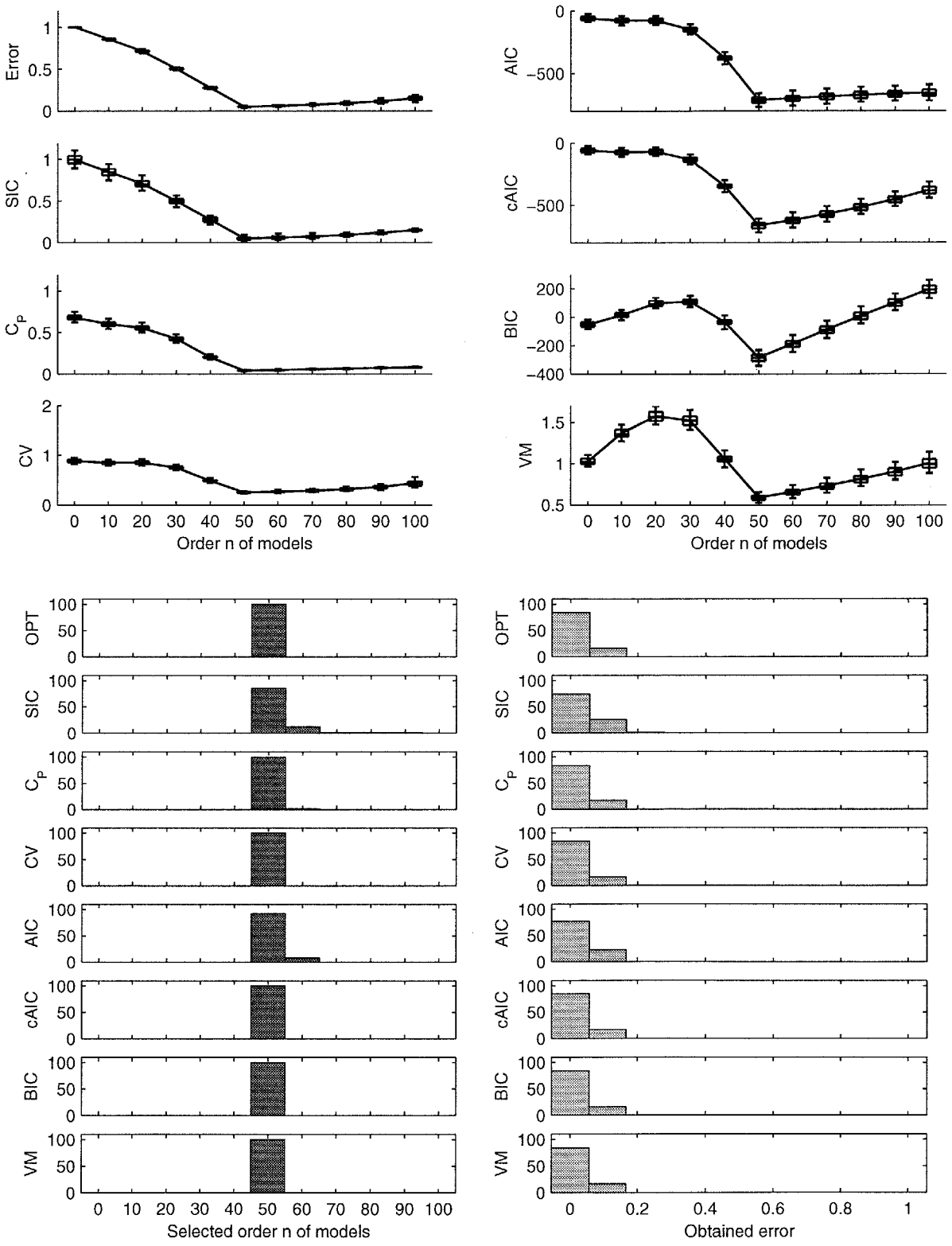


Figure 4.5: Results of LMS learning simulation when $(M, \sigma^2) = (500, 0.2)$.

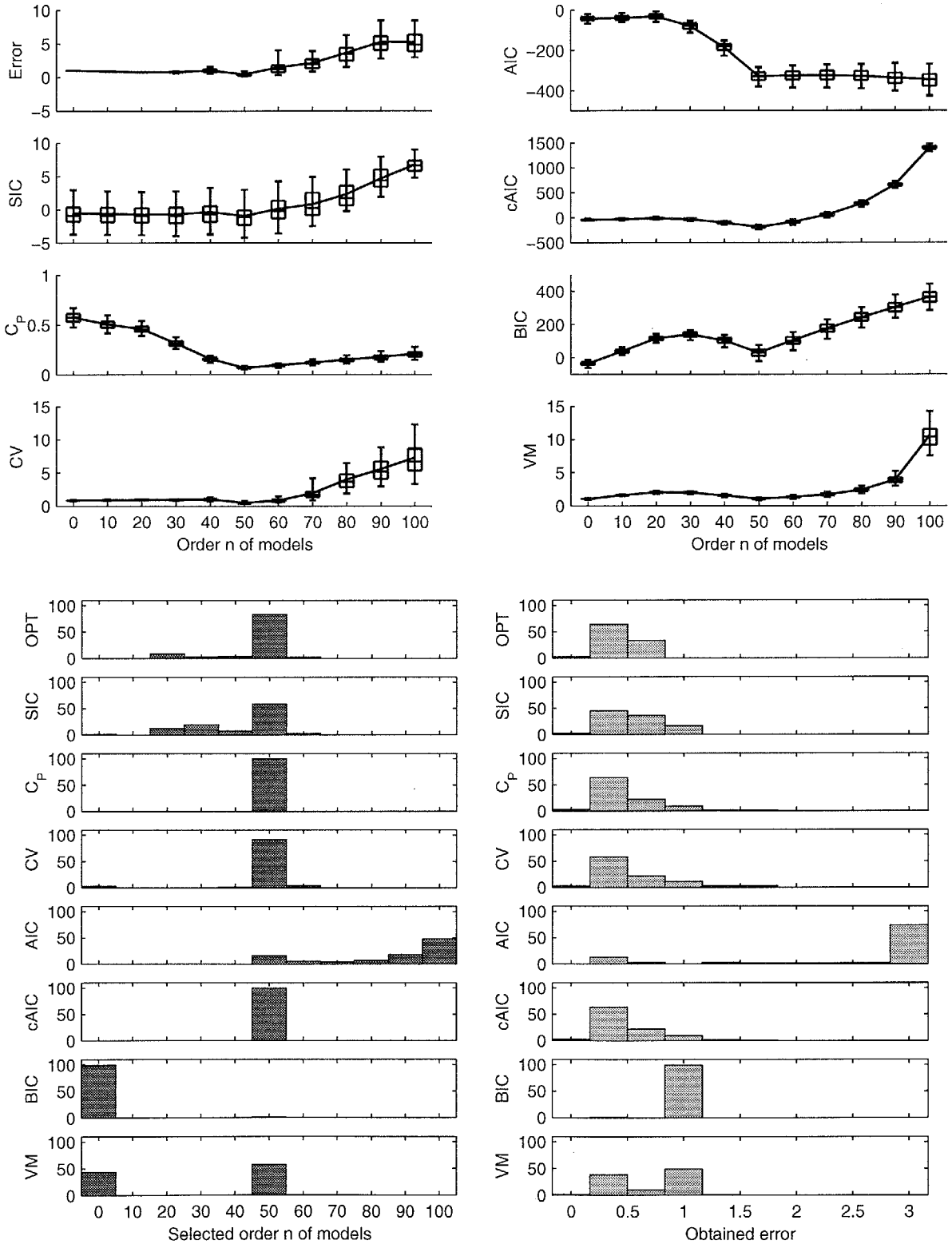


Figure 4.6: Results of LMS learning simulation when $(M, \sigma^2) = (250, 0.2)$.

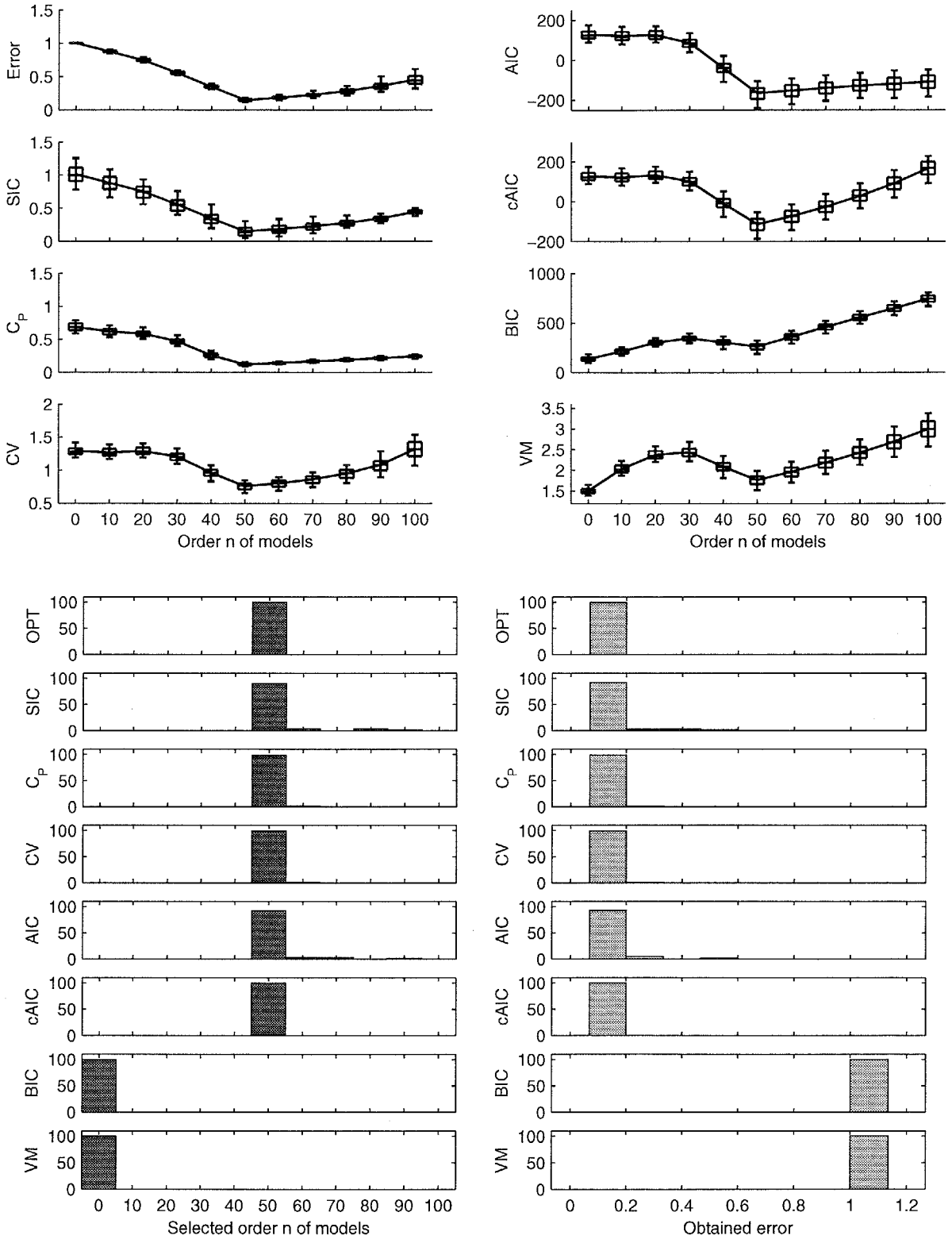


Figure 4.7: Results of LMS learning simulation when $(M, \sigma^2) = (500, 0.6)$.

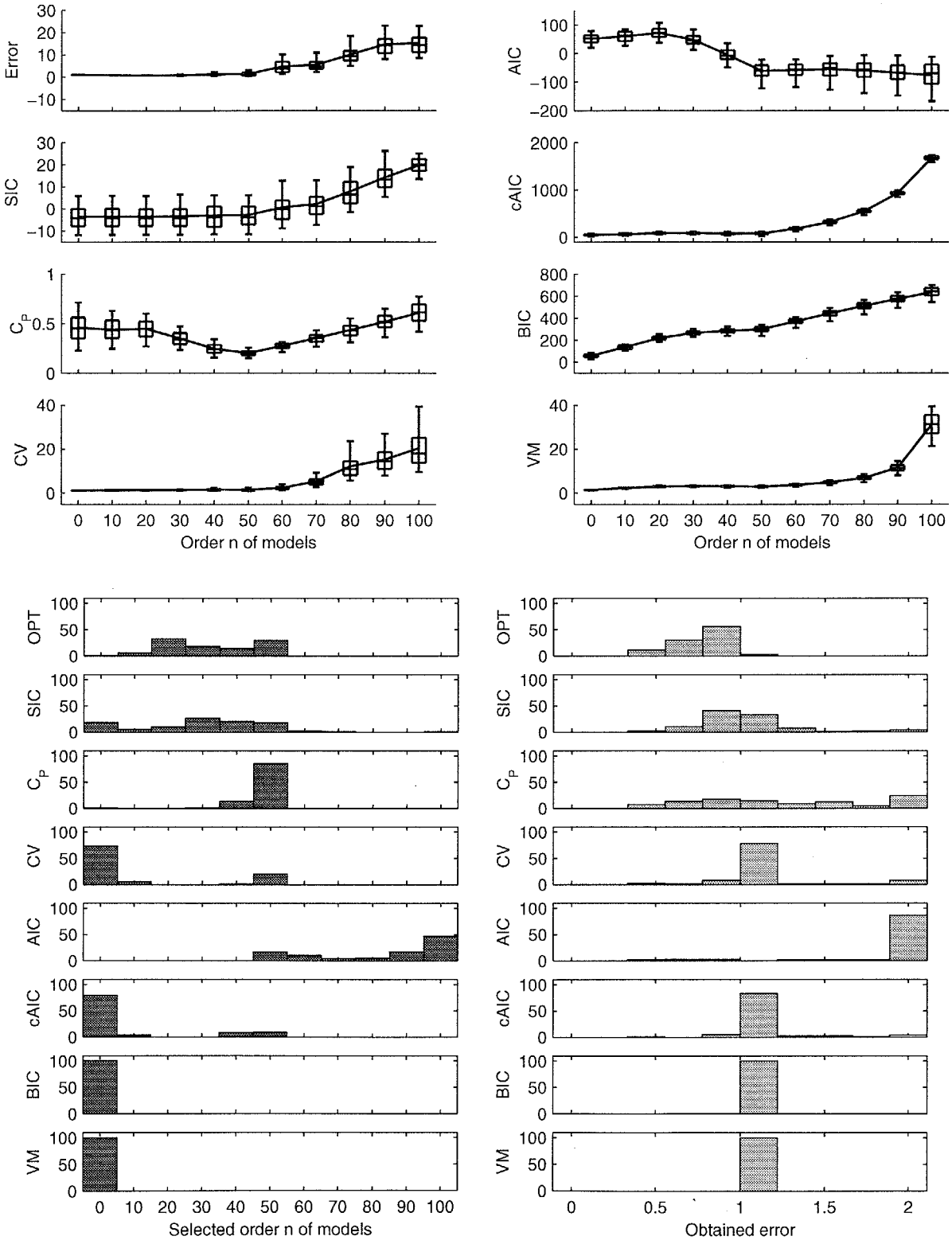


Figure 4.8: Results of LMS learning simulation when $(M, \sigma^2) = (250, 0.6)$.

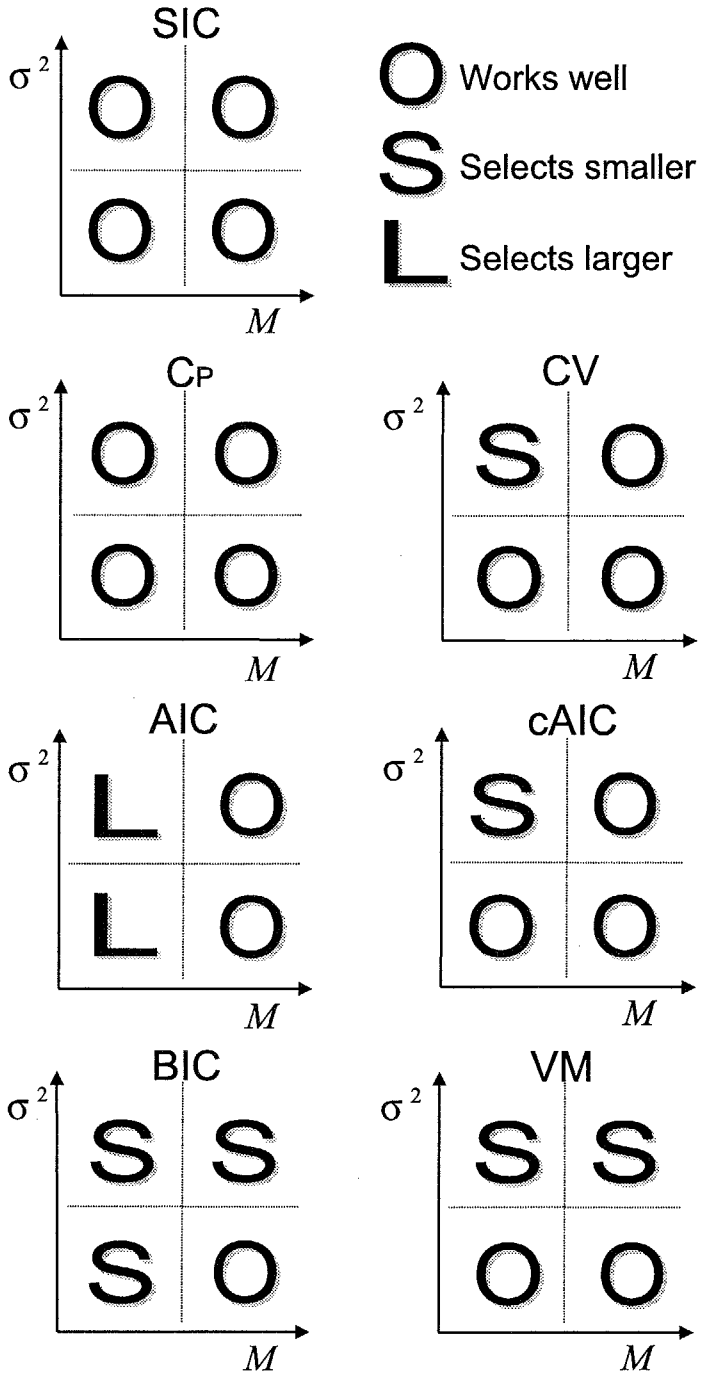


Figure 4.9: Summary of LMS learning simulations. SIC works well for any M and σ^2. Although C_P also works, it almost always specifies the true model, which is generally different from the model that provides the minimum generalization error. CV and cAIC show tendencies to select smaller models when M is small and σ^2 is large. AIC tends to select larger models when M is small. BIC is inclined to select smaller models when M is small or σ^2 is large. VM tends to select smaller models when σ^2 is large.

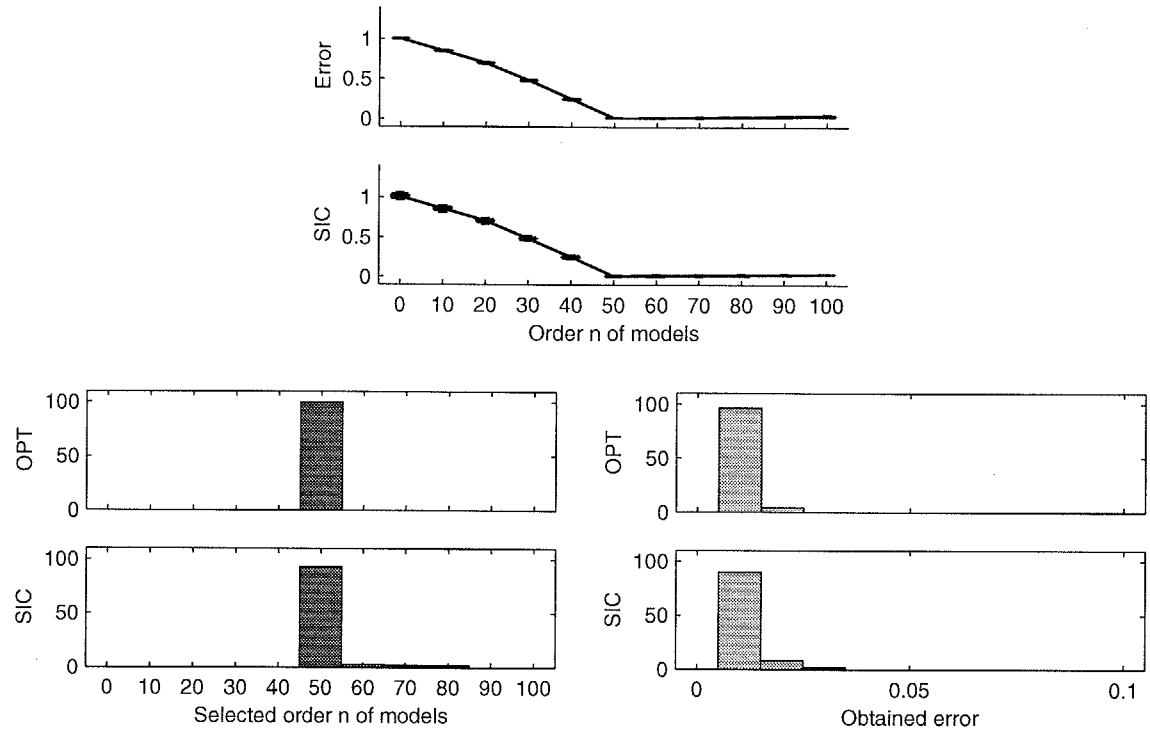


Figure 4.10: Results of LMS learning simulation with uniform noise when $M = 500$.

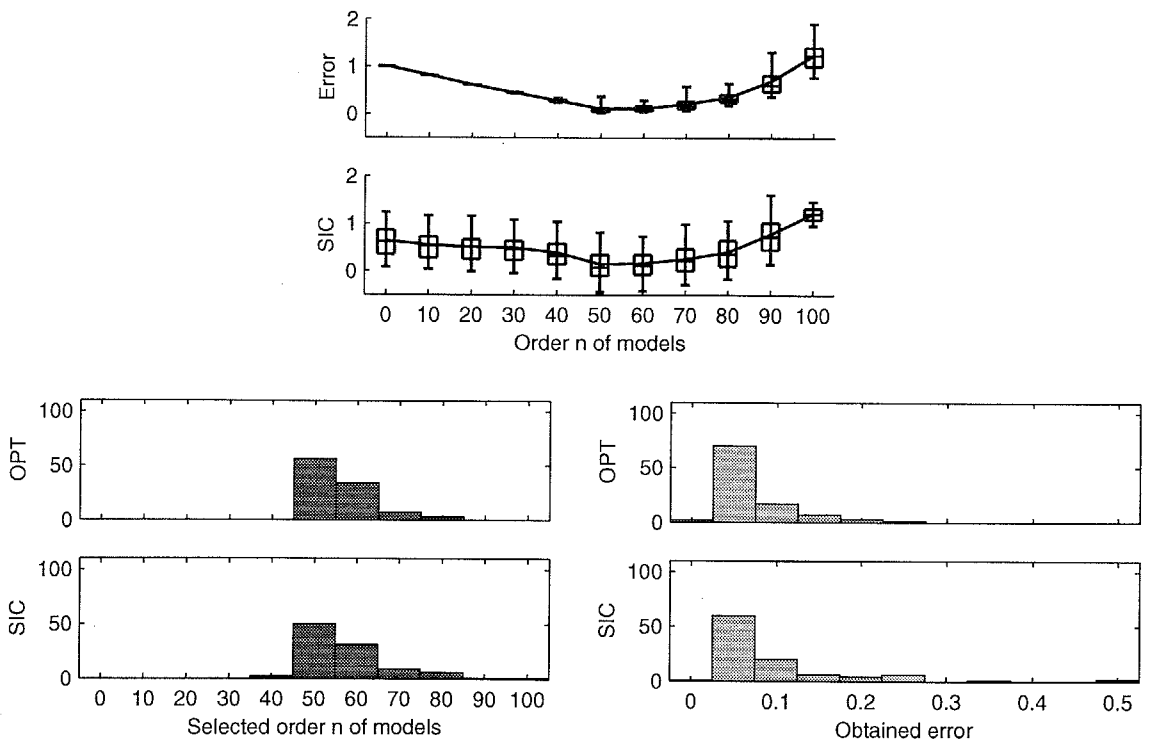


Figure 4.11: Results of LMS learning simulation with uniform noise when $M = 250$.

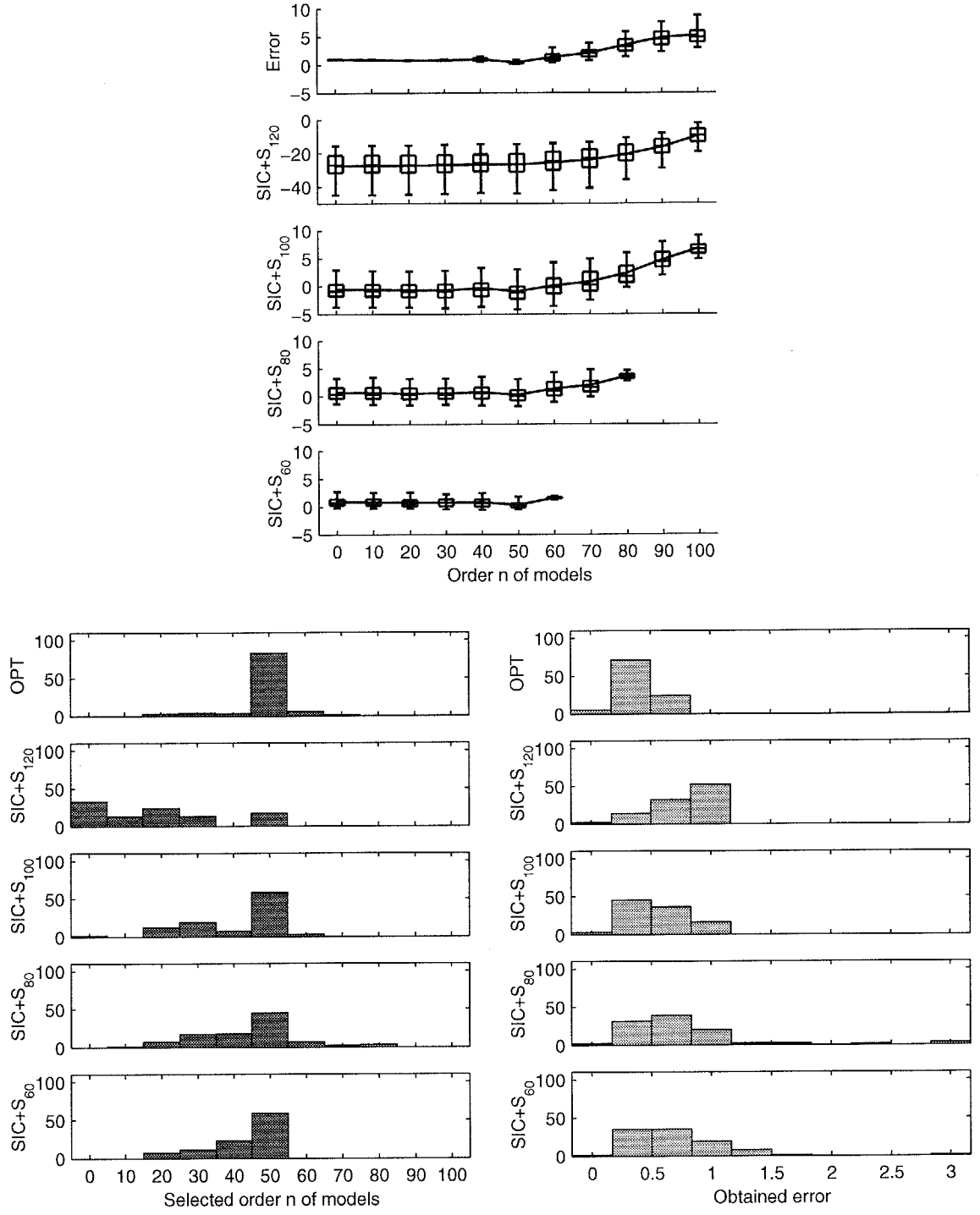


Figure 4.12: Results of LMS learning simulation with changing H for $(M, \sigma^2) = (250, 0.2)$. 'SIC+$S_n$' denotes the case when SIC is calculated with $H = S_n$.

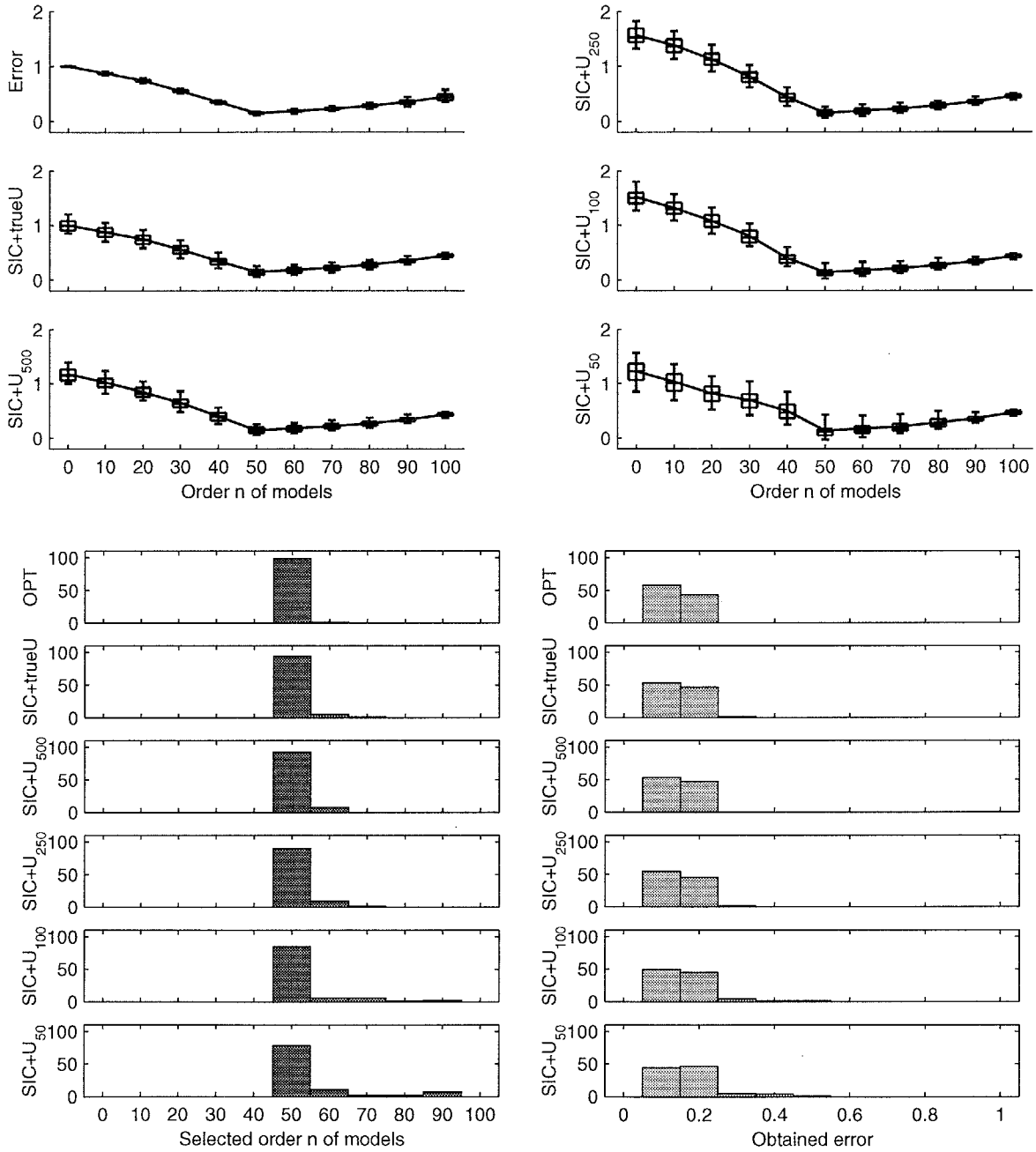


Figure 4.13: Results of LMS learning simulation with covariance matrix U estimated by using unlabeled sample points for $(M, \sigma^2) = (500, 0.6)$. 'SIC+true$U$' denotes the case when SIC is calculated with the true covariance matrix U. 'SIC+U_m' denotes the case when SIC is calculated with the covariance matrix U estimated from m unlabeled sample points.

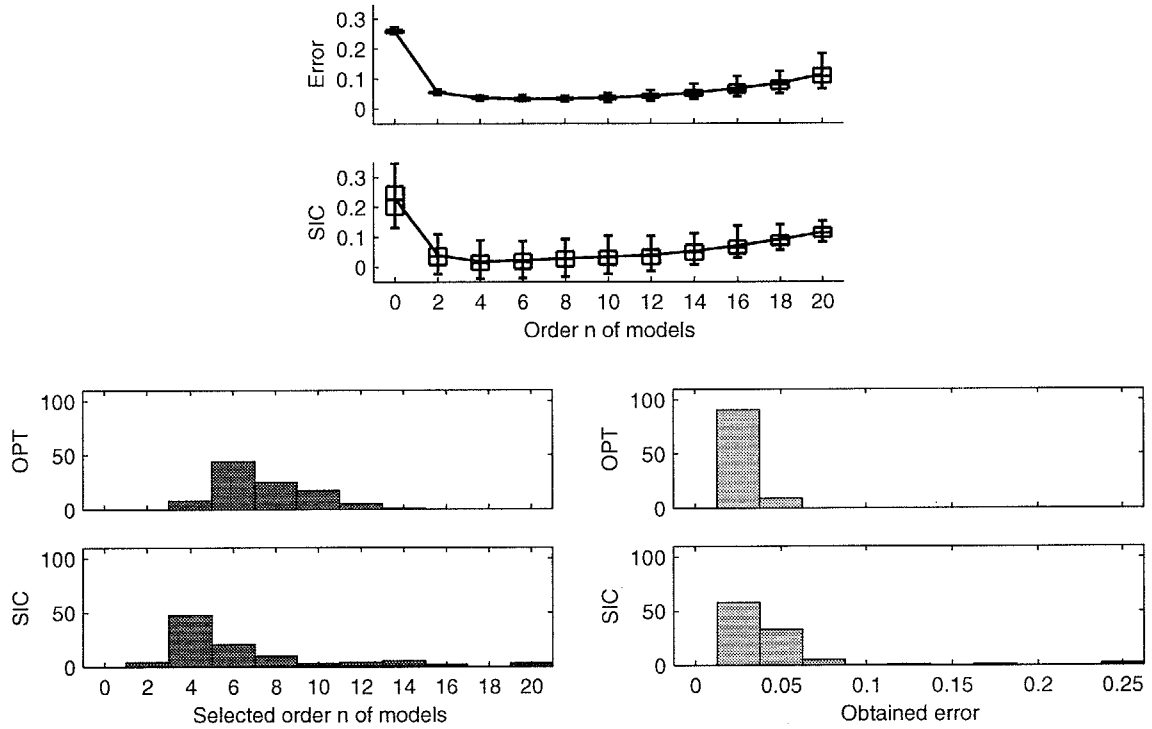


Figure 4.14: Results of LMS learning simulation with unrealizable learning target function. $f(x)$ is the step function and $(M, \sigma^2) = (100, 0.1)$.

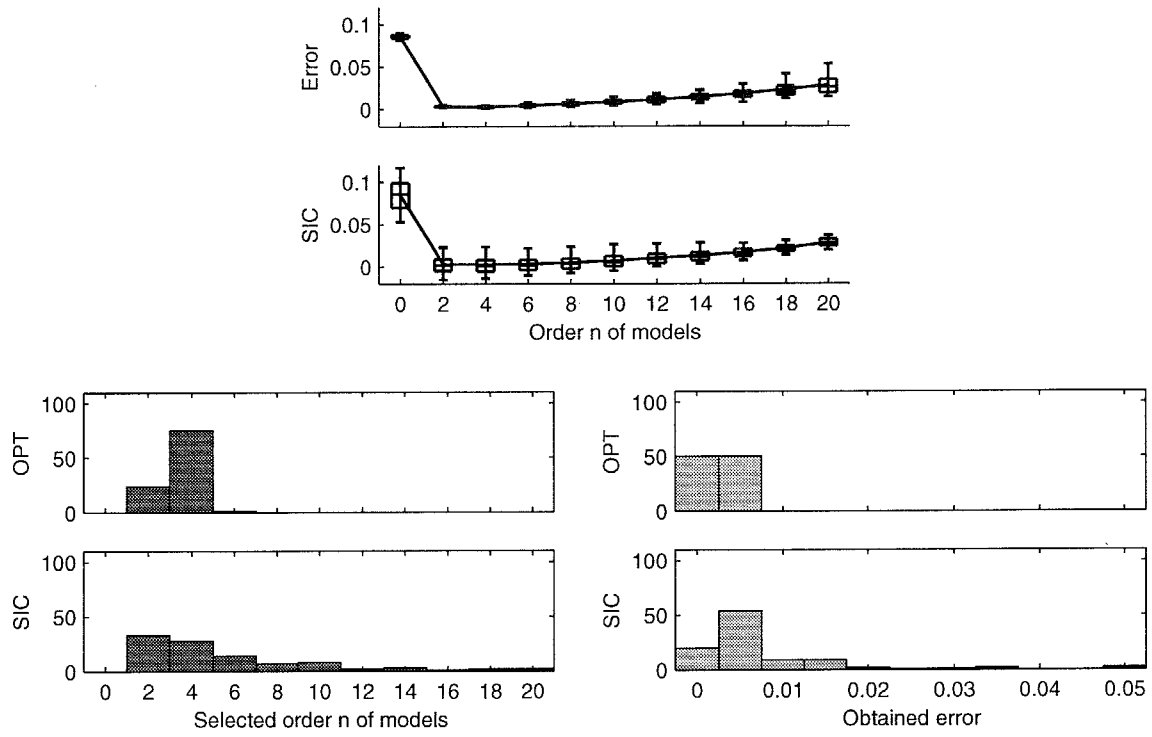


Figure 4.15: Results of LMS learning simulation with unrealizable learning target function. $f(x) = \frac{1}{1+x^2}$ and $(M, \sigma^2) = (100, 0.03)$.

4.8.2 SIC for regularization learning

SIC for regularization learning given in Section 4.5 is experimentally compared with existing model selection techniques.

4.8.2.1 Setting

Let the dimension L of the input vector $\boldsymbol{x}$ be 1, and H be spanned by the functions

$$\{x^n\}_{n=0}^{20} \tag{4.146}$$

defined on $[-1, 1]$, and the inner product is defined as

$$\langle f, g \rangle = \int_{-1}^{1} f(x)\overline{g(x)}dx, \tag{4.147}$$

i.e., H is a polynomial space of order 20 (see Section 3.3.2). Note that the dimension of H is 21.

Let the learning target function $f(x)$ be

$$f(x) = -x^2 + x^4. \tag{4.148}$$

Let the sample points $\{x_m\}_{m=1}^{M}$ be randomly created in the domain $[-1, 1]$, and the noise ϵ_m be independently subject to the same normal distribution with mean 0 and variance σ^2:

$$\epsilon_m \sim N(0, \sigma^2). \tag{4.149}$$

In this case, the noise covariance matrix Q is given as

$$Q = \sigma^2 I_M. \tag{4.150}$$

The simulation is performed 100 times for $(M, \sigma^2) = (200, 0.2)$, $(50, 0.2)$, $(200, 0.5)$, and $(50, 0.5)$, with changing the noise $\{\epsilon_m\}_{m=1}^{M}$ in each trial.

We adopt regularization learning with $T = W^{-1}$, i.e., weight decay (see Section 3.2.2). The following values are considered as candidates of the regularization parameter α:

$$\mathcal{M} = \{10^{-4}, 10^{-3}, 10^{-2}, \ldots, 10^{4}\}. \tag{4.151}$$

We shall measure the error of a learning result function $\hat{f}_{T,\alpha}(x)$ by

$$\mathrm{Error}[T, \alpha] = \|\hat{f}_{T,\alpha} - f\|^2 = \int_{-1}^{1} \left|\hat{f}_{T,\alpha}(x) - f(x)\right|^2 dx. \tag{4.152}$$

Let $\varphi_p(x) = x^p$ for $p = 0, 1, 2, \ldots, 20$. In the current setting, the covariance matrix U is given as (see Eq.(4.44))

$$[U]_{p,p'} = \int \varphi_{p'}(u)\overline{\varphi_p(u)}du, \tag{4.153}$$

where $[\,\cdot\,]_{p,p'}$ denotes the (p,p')-th element of a matrix. The Moore-Penrose generalized inverse is calculated by Eq.(4.50) with $\gamma = 0.1$.

4.8.2.2 Comparison with existing model selection methods

We compare the following model selection criteria:

(a) Subspace information criterion (SIC): X_u is obtained by Eq.(4.30) and Q is estimated by Eq.(4.31). In this case, SIC for (T, α) is given by Eq.(4.57).

(b) C_L (Mallows [73]): C_L for (T, α) is given by Eq.(4.104).

(c) Leave-one-out cross-validation (CV): A closed form expression of the leave-one-out error for (T, α) is given by Eq.(4.106).

(d) Generalized cross-validation (GCV) (Craven & Wahba [28]): GCV for (T, α) is given by Eq.(4.110).

(e) Network information criterion (NIC) (Murata *et al.* [82]): NIC for (T, α) is given by Eq.(4.86).

(f) A Bayesian information criterion (ABIC) (Akaike [2]): ABIC for α with fixed T is given by Eq.(4.120).

(g) Vapnik's measure (VM) (Cherkassky *et al.* [23]): VM for (T, α) is given by Eqs.(4.96) and (4.99).

Figures 4.16, 4.17, 4.18, and 4.19 display the simulation results for $(M, \sigma^2) = (200, 0.2)$, $(50, 0.2)$, $(200, 0.5)$, and $(50, 0.5)$, respectively. The top eight graphs show the values of the error and model selection criteria corresponding to the regularization parameter α in log-scale (see Eq.(4.151)). The box plot notation specifies marks at 95, 75, 50, 25, and 5 percentiles of values. The solid line denotes the mean values. The bottom-left eight graphs show the distributions of the selected regularization parameter in log-scale. 'OPT' indicates the optimal regularization parameter that minimizes the error measured

by Eq.(4.152). The bottom-right eight graphs show the distributions of the error obtained by the regularization parameter selected by each criterion.

When $(M, \sigma^2) = (200, 0.2)$ (Figure 4.16), SIC, C_L, CV, GCV, ABIC, and VM almost always select good regularization parameters. Although NIC is inclined to select smaller regularization parameters, all the criteria provide almost the optimal generalization capability.

When $(M, \sigma^2) = (50, 0.2)$ (Figure 4.17), SIC, C_L, CV, GCV, and ABIC almost always select good regularization parameters. In contrast, NIC shows a tendency to select smaller regularization parameters, and VM tends to select larger regularization parameters. As a result, VM gives larger errors.

When $(M, \sigma^2) = (200, 0.5)$ (Figure 4.18), the results are almost the same as Figure 4.17.

Finally, when $(M, \sigma^2) = (50, 0.5)$ (Figure 4.19), SIC, C_L, CV, GCV, and ABIC almost always select good regularization parameters. In contrast, NIC shows a tendency to select smaller regularization parameters, and VM tends to select larger regularization parameters. As a result, NIC and VM provide larger errors.

This simulation shows that SIC gives a good estimate of the error on average, and it works as well as C_L, CV, GCV, and ABIC.

4.8.2.3 Active design of optimal regularization parameter

Now we experimentally evaluate the regularization parameter $\alpha_{\widehat{SIC}}$ given in Section 4.5.3.1. The simulation is performed with the same setting as Section 4.8.2.1.

Figures 4.20, 4.21, 4.22, and 4.23 display the simulation results for $(M, \sigma^2) = (200, 0.2)$, $(50, 0.2)$, $(200, 0.5)$, and $(50, 0.5)$, respectively. The left three graphs show the distributions of α_{OPT}, α_{SIC}, and $\alpha_{\widehat{SIC}}$ in log-scale. Here, α_{OPT} is the minimizer of the error measured by Eq.(4.152), α_{SIC} is the minimizer of SIC, and $\alpha_{\widehat{SIC}}$ is the minimizer of $\widehat{\mathrm{SIC}}$ (see Section 4.5.3.1). In the simulation, α_{OPT} and α_{SIC} are searched from 10^{-4} to 10^{4} in the scale of 10^{-4}. The middle three graphs show the distributions of Error$[\alpha_{OPT}]$, Error$[\alpha_{SIC}]$, and Error$[\alpha_{\widehat{SIC}}]$. The right two graphs show the distributions of SIC$[\alpha_{SIC}]$ and SIC$[\alpha_{\widehat{SIC}}]$.

When $M = 200$ (Figures 4.20 and 4.22), $\alpha_{\widehat{SIC}}$ almost minimizes SIC (see the right two graphs). In this case, Error$[\alpha_{\widehat{SIC}}]$ is almost the same as Error$[\alpha_{OPT}]$ (see the middle three graphs). When $M = 50$ (Figure 4.21 and 4.23), SIC$[\alpha_{\widehat{SIC}}]$ is slightly larger than

SIC[α_{SIC}]. However, the middle three graphs show that Error[$\alpha_{\widehat{SIC}}$] is almost the same as Error[α_{OPT}].

The simulations show that $\alpha_{\widehat{SIC}}$ gives higher levels of the generalization capability when $\alpha_{\widehat{SIC}}$ almost minimizes SIC (Figures 4.20 and 4.22). Moreover, even when $\alpha_{\widehat{SIC}}$ is not a very good estimate of α_{SIC}, $\alpha_{\widehat{SIC}}$ provides higher levels of the generalization capability (Figure 4.21 and 4.23). It should be noted that in any cases, α_{SIC} is slightly larger than α_{OPT}, and $\alpha_{\widehat{SIC}}$ is slightly smaller than α_{SIC}. As a result, Error[$\alpha_{\widehat{SIC}}$] is slightly better than Error[α_{OPT}].

In the simulations, we used Tikhonov's regularization for calculating the Moore-Penrose generalized inverse. Thanks to this, the value of $\lambda_{\max}$ (see Section 4.5.3) is kept small. This implies that the reliability of $\alpha_{\widehat{SIC}}$ can be improved by Tikhonov's regularization.

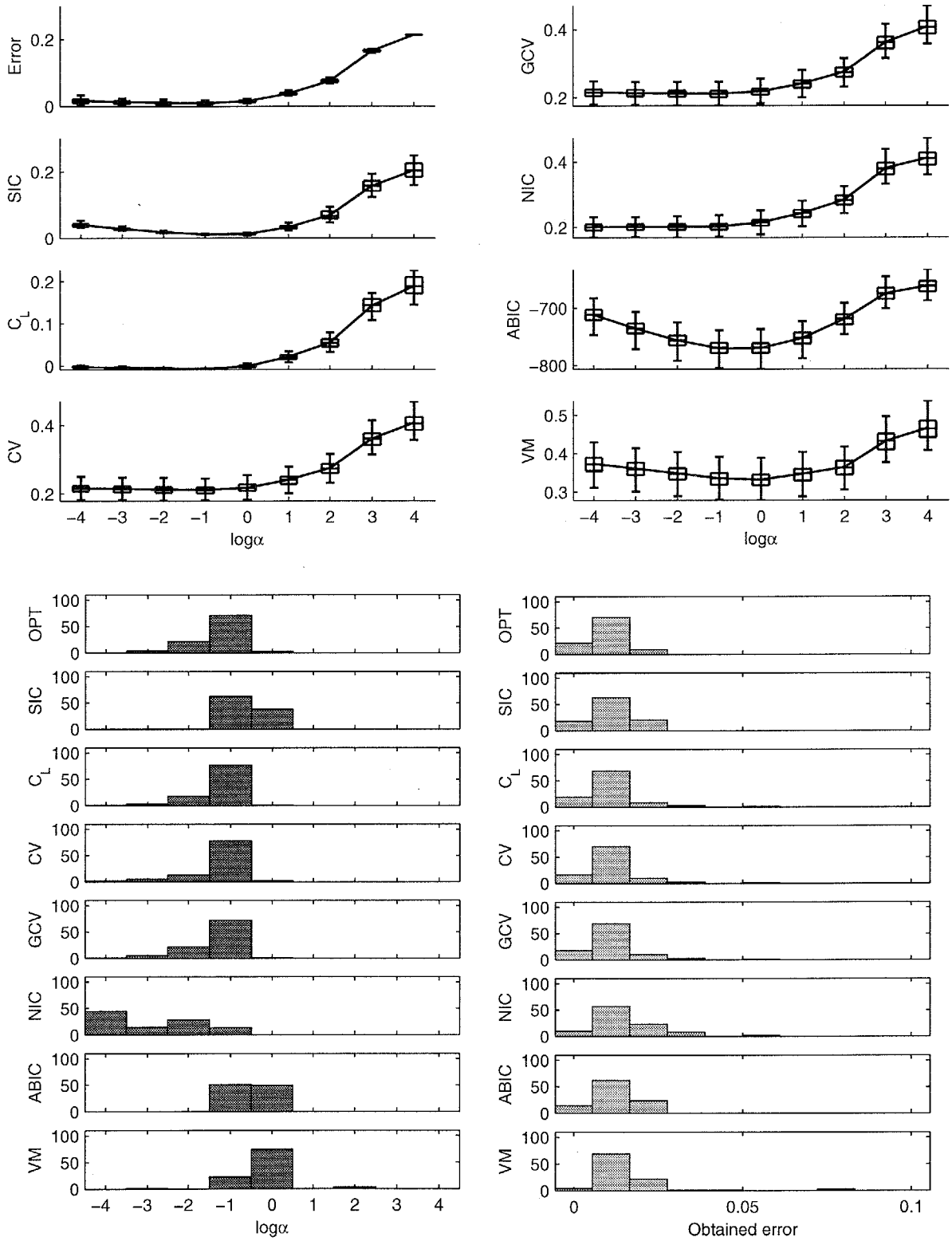


Figure 4.16: Results of regularization learning simulation when $(M, \sigma^2) = (200, 0.2)$.

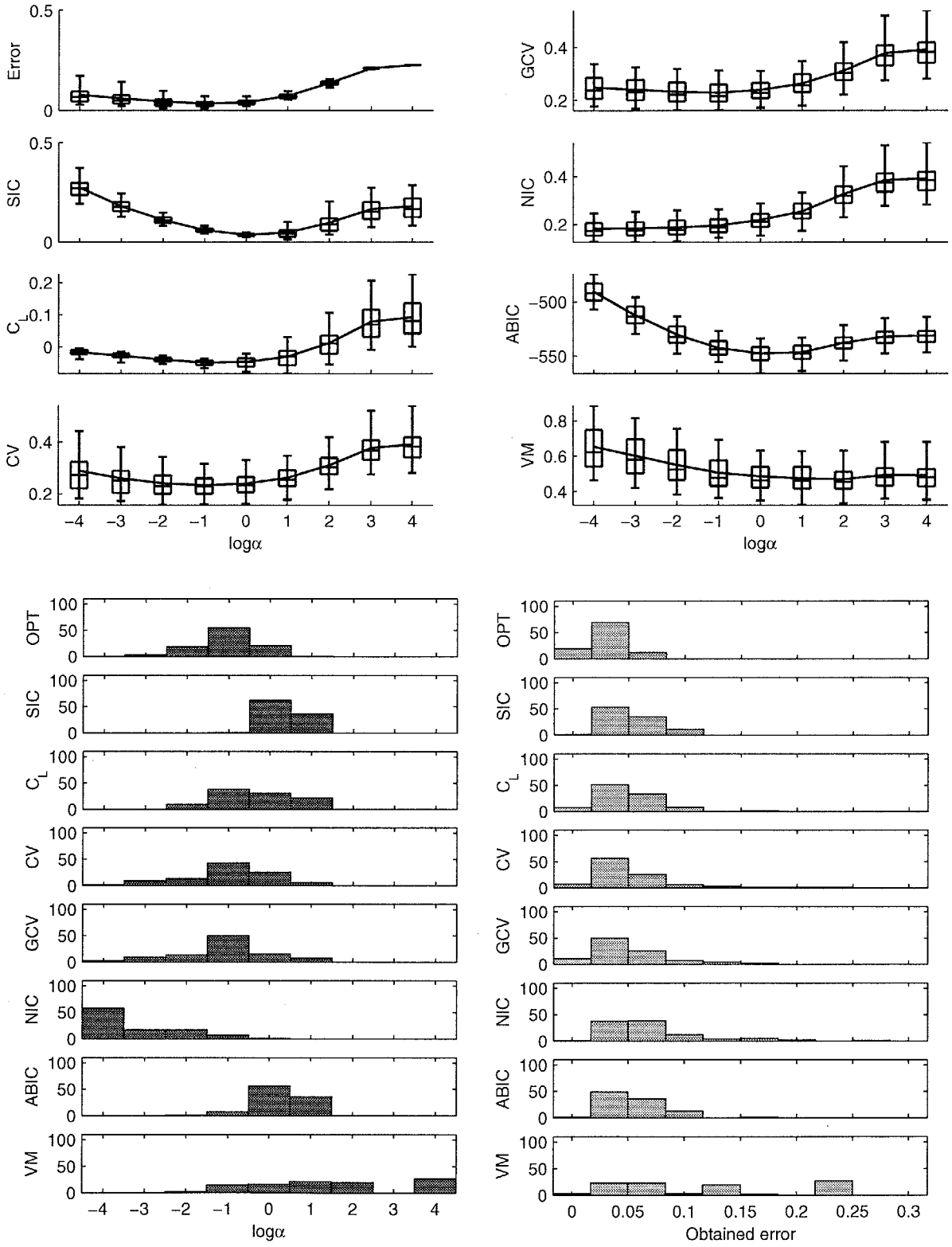


Figure 4.17: Results of regularization learning simulation when $(M, \sigma^2) = (50, 0.2)$.

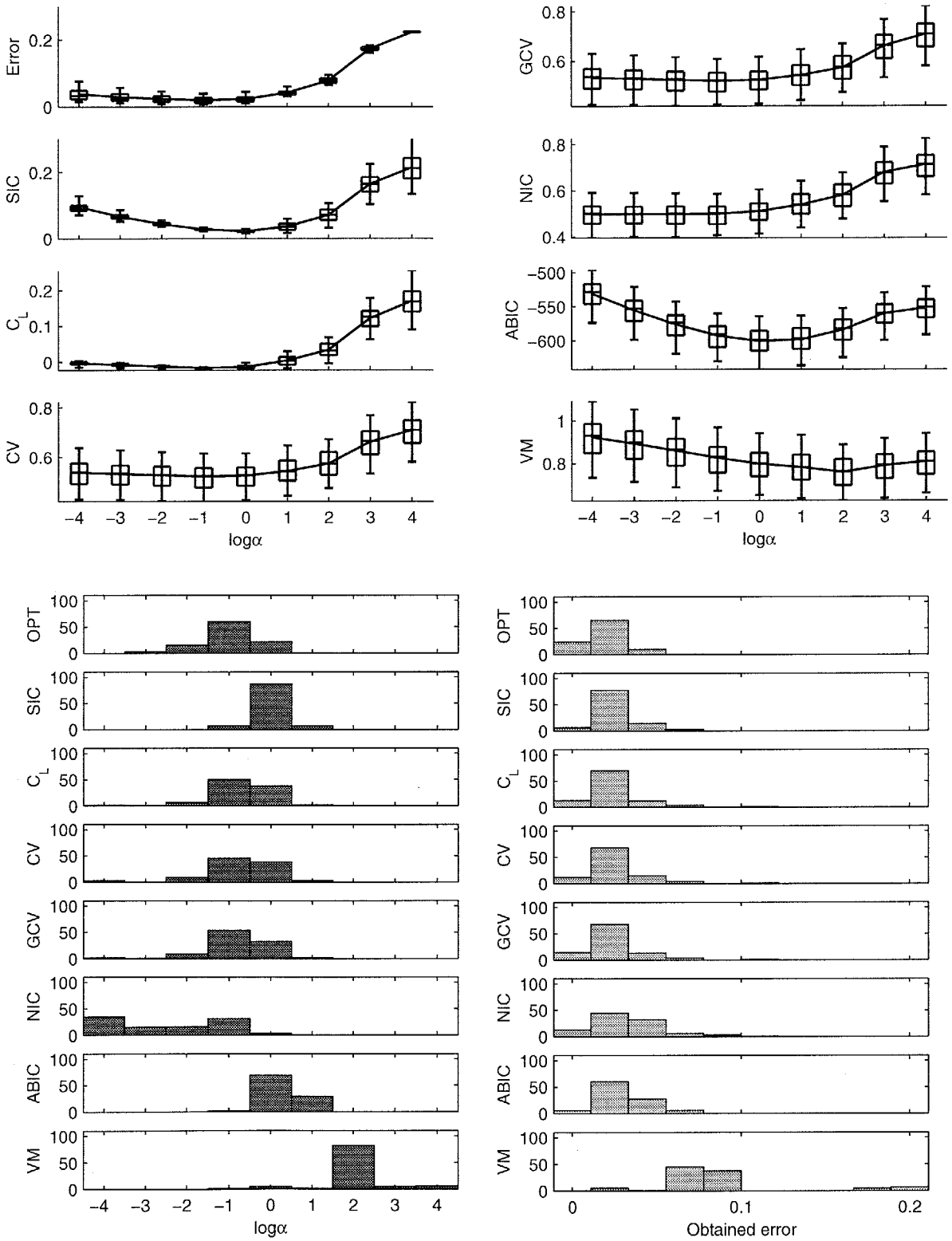


Figure 4.18: Results of regularization learning simulation when $(M, \sigma^2) = (200, 0.5)$.

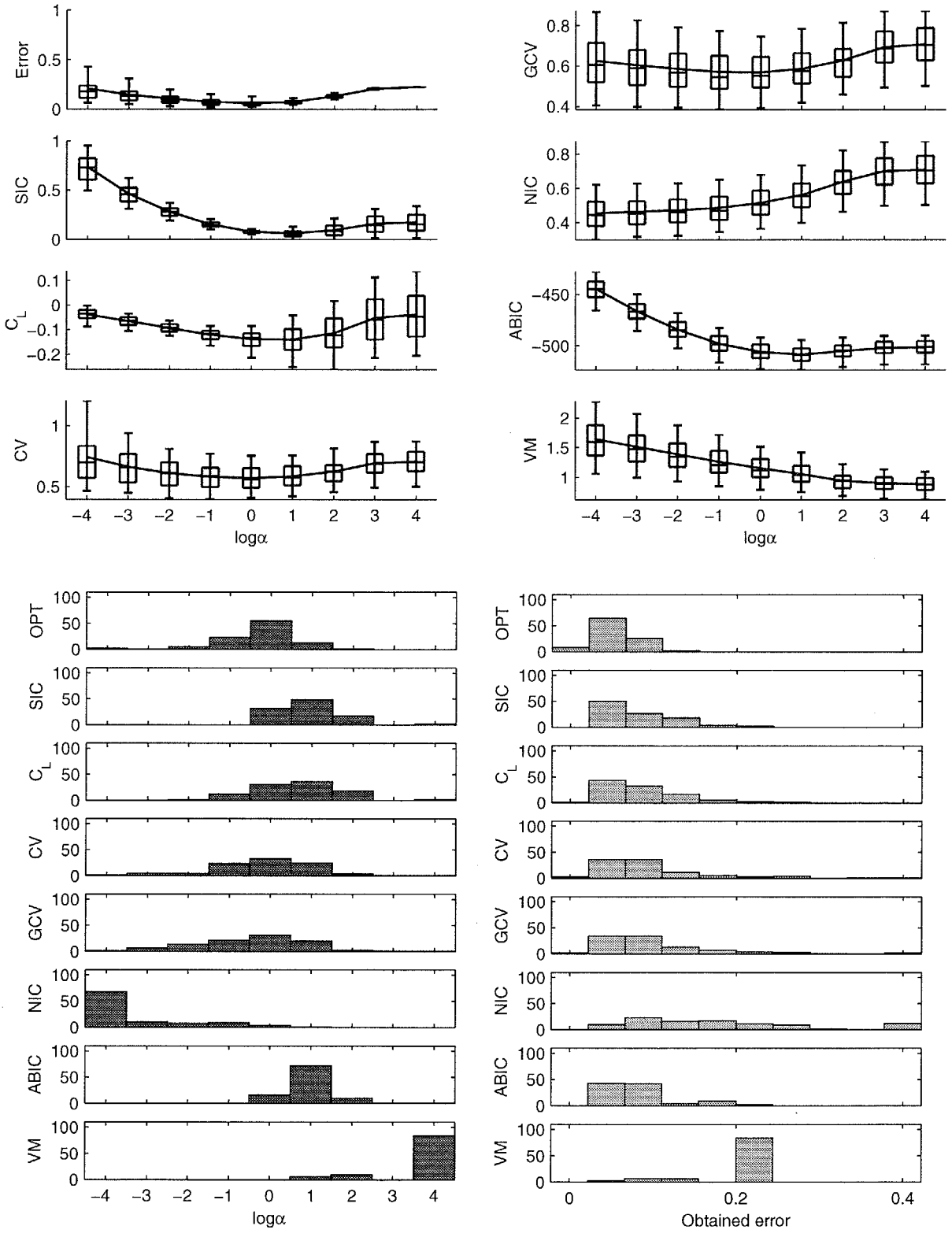


Figure 4.19: Results of regularization learning simulation when $(M, \sigma^2) = (50, 0.5)$.

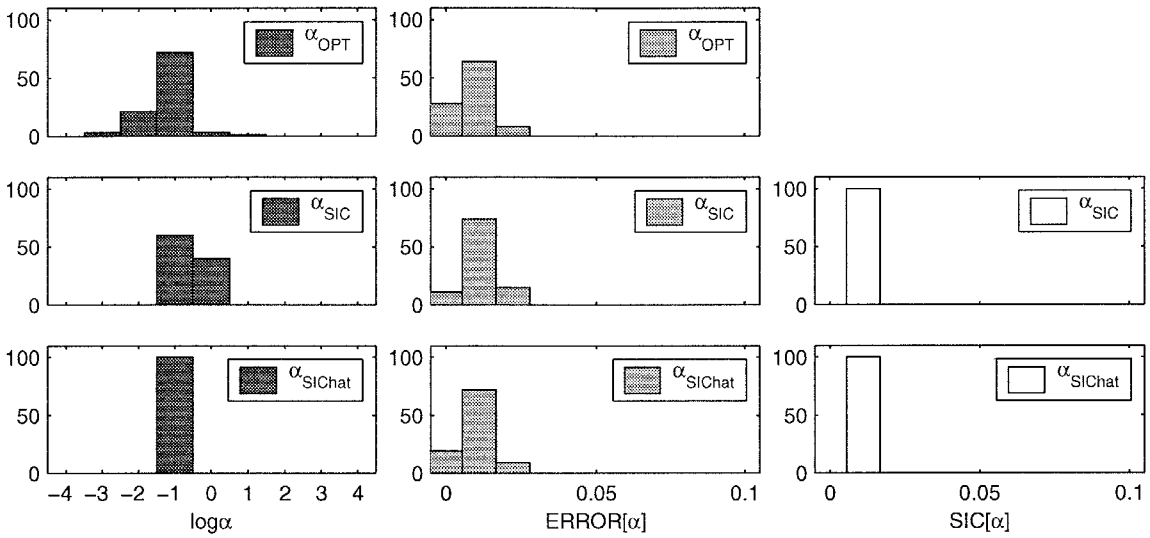


Figure 4.20: Results of regularization learning simulation for $\alpha_{\widehat{SIC}}$ when $(M, \sigma^2) = (200, 0.2)$.

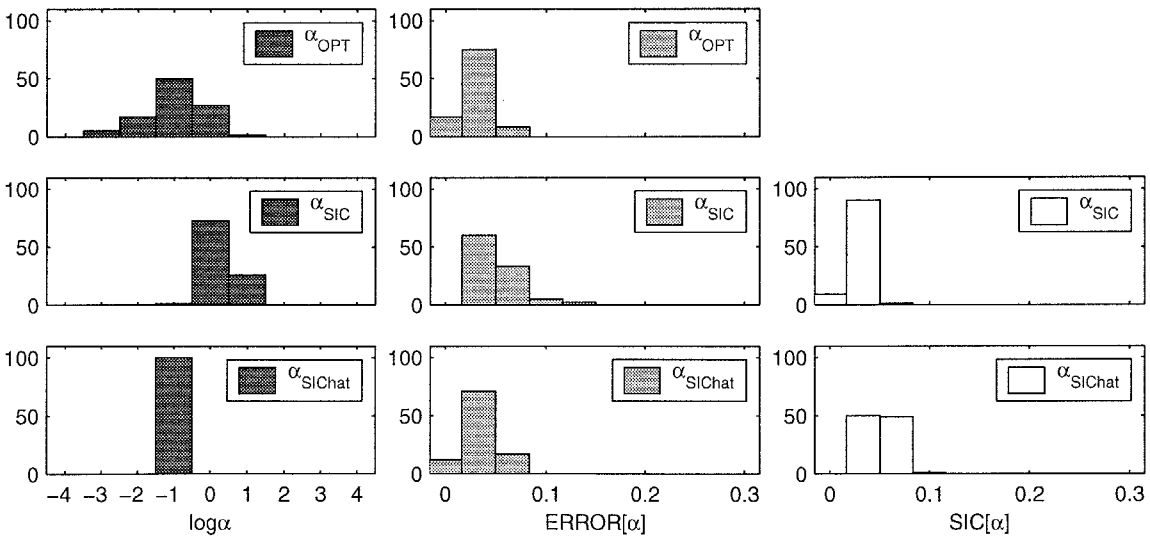


Figure 4.21: Results of regularization learning simulation for $\alpha_{\widehat{SIC}}$ when $(M, \sigma^2) = (50, 0.2)$.

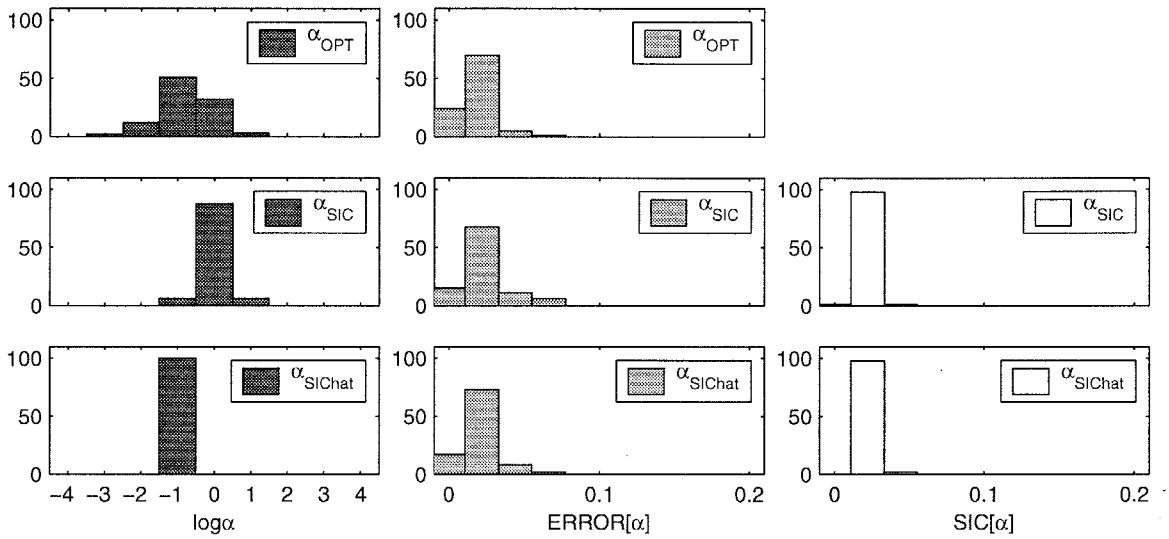


Figure 4.22: Results of regularization learning simulation for $\alpha_{\widehat{SIC}}$ when $(M, \sigma^2) = (200, 0.5)$.

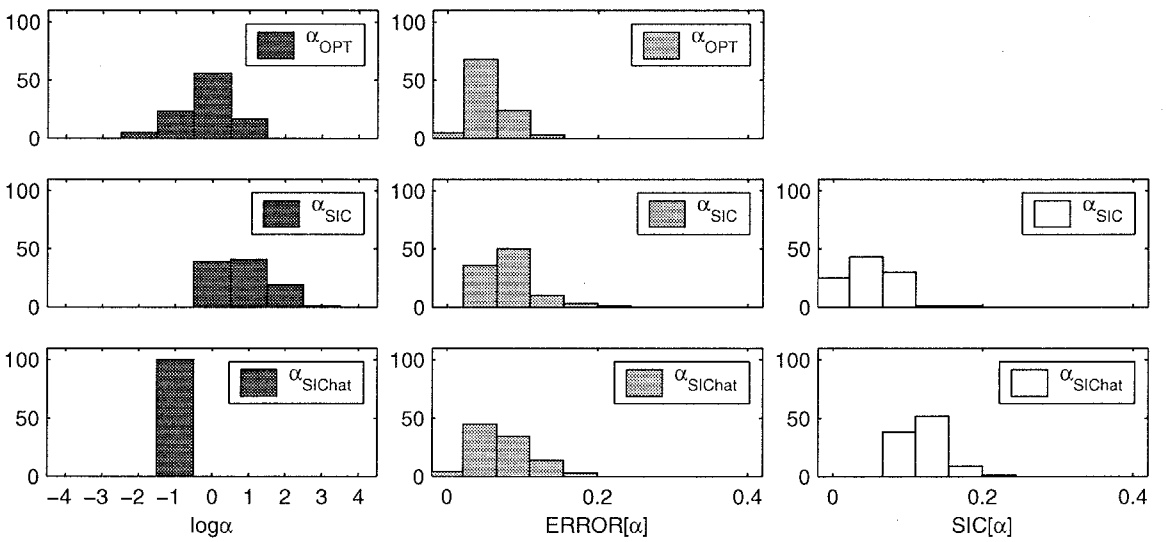


Figure 4.23: Results of regularization learning simulation for $\alpha_{\widehat{SIC}}$ when $(M, \sigma^2) = (50, 0.5)$.

4.9 Proofs

In this section, proofs of all theorems, corollaries, and lemmas given in this chapter are provided.

4.9.1 Lemma 4.3

It follows from Eqs.(4.18), (4.5), (4.8), (4.13), (4.11), (4.7), (4.16), (4.17), (4.9), and (4.14) that

$$\begin{aligned}
\mathrm{E}_{\boldsymbol{\epsilon}}\mathrm{SIC} &= \mathrm{E}_{\boldsymbol{\epsilon}}\left(\|(X_\theta - X_u)\boldsymbol{y}\|^2 - \mathrm{tr}(X_\theta - X_u)Q(X_\theta - X_u)^* + \mathrm{tr}X_\theta Q X_\theta^*\right) \\
&= \mathrm{E}_{\boldsymbol{\epsilon}}\left(\|\hat{f}_\theta - \hat{f}_u\|^2\right) - \mathrm{tr}(X_\theta - X_u)Q(X_\theta - X_u)^* + \mathrm{tr}X_\theta Q X_\theta^* \\
&= \mathrm{E}_{\boldsymbol{\epsilon}}\left(\|\mathrm{E}_{\boldsymbol{\epsilon}}(\hat{f}_\theta - \hat{f}_u) - \mathrm{E}_{\boldsymbol{\epsilon}}(\hat{f}_\theta - \hat{f}_u) + \hat{f}_\theta - \hat{f}_u\|^2\right) \\
&\quad - \mathrm{tr}(X_\theta - X_u)Q(X_\theta - X_u)^* + \mathrm{E}_{\boldsymbol{\epsilon}}\|X_\theta\boldsymbol{\epsilon}\|^2 \\
&= \|\mathrm{E}_{\boldsymbol{\epsilon}}(\hat{f}_\theta - \hat{f}_u)\|^2 - 2\mathrm{E}_{\boldsymbol{\epsilon}}\mathrm{Re}\langle \mathrm{E}_{\boldsymbol{\epsilon}}(\hat{f}_\theta - \hat{f}_u), \mathrm{E}_{\boldsymbol{\epsilon}}(\hat{f}_\theta - \hat{f}_u) - (\hat{f}_\theta - \hat{f}_u)\rangle \\
&\quad + \mathrm{E}_{\boldsymbol{\epsilon}}\|\mathrm{E}_{\boldsymbol{\epsilon}}(\hat{f}_\theta - \hat{f}_u) - (\hat{f}_\theta - \hat{f}_u)\|^2 - \mathrm{tr}(X_\theta - X_u)Q(X_\theta - X_u)^* + \mathrm{E}_{\boldsymbol{\epsilon}}\|X_\theta\boldsymbol{y} - X_\theta\boldsymbol{z}\|^2 \\
&= \|\mathrm{E}_{\boldsymbol{\epsilon}}\hat{f}_\theta - f\|^2 + \mathrm{tr}(X_\theta - X_u)Q(X_\theta - X_u)^* \\
&\quad - \mathrm{tr}(X_\theta - X_u)Q(X_\theta - X_u)^* + \mathrm{E}_{\boldsymbol{\epsilon}}\|X_\theta\boldsymbol{y} - \mathrm{E}_{\boldsymbol{\epsilon}}X_\theta\boldsymbol{y}\|^2 \\
&= \|\mathrm{E}_{\boldsymbol{\epsilon}}\hat{f}_\theta - f\|^2 + \mathrm{E}_{\boldsymbol{\epsilon}}\|\hat{f}_\theta - \mathrm{E}_{\boldsymbol{\epsilon}}\hat{f}_\theta\|^2 \\
&= \mathrm{E}_{\boldsymbol{\epsilon}}\|\hat{f}_\theta - f\|^2, \qquad (4.154)
\end{aligned}$$

which concludes the proof. ■

4.9.2 Corollary 4.4

It follows from Eqs.(4.25), (4.42), (4.27), and (4.40) that

$$\begin{aligned}
AW &= \sum_{m=1}^{M}\left(\boldsymbol{e}_m \otimes \overline{K(\cdot, \boldsymbol{x}_m)}\right)\sum_{p=1}^{\mu}(\varphi_p \otimes \overline{\boldsymbol{e}_p}) \\
&= \sum_{m=1}^{M}\langle\varphi_p, K(\cdot, \boldsymbol{x}_m)\rangle\,(\boldsymbol{e}_m \otimes \overline{\boldsymbol{e}_p}) \\
&= \sum_{m=1}^{M}\varphi_p(\boldsymbol{x}_m)\,(\boldsymbol{e}_m \otimes \overline{\boldsymbol{e}_p}) \\
&= B. \qquad (4.155)
\end{aligned}$$

Operating W^{-1} from the right-hand side of Eq.(4.155), we have

$$A = BW^{-1}. \tag{4.156}$$

In this case, it follows from Theorem 4.11 in Albert [5] that

$$A^\dagger = WB^\dagger. \tag{4.157}$$

Similarly, it holds that

$$A_S^\dagger = WB_S^\dagger. \tag{4.158}$$

It follows from Eqs.(4.42) and (4.44) that

$$\begin{aligned} W^*W &= \sum_{p=1}^{\mu}\sum_{p'=1}^{\mu}(\boldsymbol{e}_p \otimes \overline{\varphi_p})(\varphi_{p'} \otimes \overline{\boldsymbol{e}_{p'}}) = \sum_{p=1}^{\mu}\sum_{p'=1}^{\mu}\langle\varphi_{p'}, \varphi_p\rangle(\boldsymbol{e}_p \otimes \overline{\boldsymbol{e}_{p'}}) \\ &= U. \end{aligned} \tag{4.159}$$

Substituting Eqs.(4.158), (4.157), and (4.159) into Eq.(4.38), we have

$$\begin{aligned} \mathrm{SIC}[S] = &\langle U(B_S^\dagger - B^\dagger)\boldsymbol{y}, (B_S^\dagger - B^\dagger)\boldsymbol{y}\rangle - \hat{\sigma}^2 \mathrm{tr} U(B_S^\dagger - B^\dagger)(B_S^\dagger - B^\dagger)^* \\ &+ \hat{\sigma}^2 \mathrm{tr} U B_S^\dagger (B_S^\dagger)^*, \end{aligned} \tag{4.160}$$

which implies Eq.(4.45). It follows from Eqs.(4.31), (4.26), (4.6), (4.8), (4.30), (4.156), and (4.157) that

$$\begin{aligned} \hat{\sigma}^2 &= \frac{\sum_{m=1}^{M}\left|\hat{f}_u(\boldsymbol{x}_m) - y_m\right|^2}{M-\mu} = \frac{\|A\hat{f}_u - \boldsymbol{y}\|^2}{M-\mu} \\ &= \frac{\|AA^\dagger\boldsymbol{y} - \boldsymbol{y}\|^2}{M-\mu} = \frac{\|BW^{-1}WB^\dagger\boldsymbol{y} - \boldsymbol{y}\|^2}{M-\mu} \\ &= \frac{\|BB^\dagger\boldsymbol{y} - \boldsymbol{y}\|^2}{M-\mu}, \end{aligned} \tag{4.161}$$

which implies Eq.(4.46). ■

4.9.3 Corollary 4.5

From Eqs.(4.36), (4.37), (4.158), and (4.42), the learning result function $\hat{f}_S(\boldsymbol{x})$ is given as

$$\begin{aligned} \hat{f}_S(\boldsymbol{x}) &= X_S\boldsymbol{y} = A_S^\dagger\boldsymbol{y} = WB_S^\dagger\boldsymbol{y} \\ &= \sum_{p=1}^{\mu}\langle B_S^\dagger\boldsymbol{y}, \boldsymbol{e}_p\rangle\varphi_p(\boldsymbol{x}), \end{aligned} \tag{4.162}$$

which implies Eq.(4.47). ■

4.9.4 Corollary 4.6

It follows from Eqs.(4.54), (4.156), and (4.56) that

$$\begin{aligned} X_{T,\alpha} &= (A^*A + \alpha T^*T)^{-1}A^* \\ &= ((W^{-1})^*B^*BW^{-1} + \alpha(W^{-1})^*W^*T^*TWW^{-1})^{-1}(W^{-1})^*B^* \\ &= ((W^{-1})^*D_{T,\alpha}W^{-1})^{-1}(W^{-1})^*B^* \\ &= WD_{T,\alpha}^{-1}W^*(W^{-1})^*B^* \\ &= WD_{T,\alpha}^{-1}B^*. \end{aligned} \tag{4.163}$$

Substituting Eqs.(4.163), (4.157), and (4.159) into Eq.(4.55), we have

$$\begin{aligned} \mathrm{SIC}[T,\alpha] &= \|W(D_{T,\alpha}^{-1}B^* - B^\dagger)\boldsymbol{y}\|^2 - \hat{\sigma}^2\mathrm{tr}W(D_{T,\alpha}^{-1}B^* - B^\dagger)(BD_{T,\alpha}^{-1} - (B^\dagger)^*)W^* \\ &\quad + \hat{\sigma}^2\mathrm{tr}WD_{T,\alpha}^{-1}B^*BD_{T,\alpha}^{-1}W^* \\ &= \langle U(D_{T,\alpha}^{-1}B^* - B^\dagger)\boldsymbol{y}, (D_{T,\alpha}^{-1}B^* - B^\dagger)\boldsymbol{y}\rangle \\ &\quad - \hat{\sigma}^2\mathrm{tr}U(D_{T,\alpha}^{-1}B^*BD_{T,\alpha}^{-1} - D_{T,\alpha}^{-1}B^*(B^\dagger)^* - B^\dagger BD_{T,\alpha}^{-1} + B^\dagger(B^\dagger)^*) \\ &\quad + \hat{\sigma}^2\mathrm{tr}UD_{T,\alpha}^{-1}B^*BD_{T,\alpha}^{-1}, \end{aligned} \tag{4.164}$$

which implies Eq.(4.57). ■

4.9.5 Corollary 4.7

From Eqs.(4.53), (4.163), and (4.42), the learning result function $\hat{f}_{T,\alpha}(\boldsymbol{x})$ is given as

$$\begin{aligned} \hat{f}_{T,\alpha}(\boldsymbol{x}) &= X_{T,\alpha}\boldsymbol{y} = WD_{T,\alpha}^{-1}B^*\boldsymbol{y} \\ &= \sum_{p=1}^{\mu}\langle D_{T,\alpha}^{-1}B^*\boldsymbol{y}, \boldsymbol{e}_p\rangle\varphi_p(\boldsymbol{x}), \end{aligned} \tag{4.165}$$

which implies Eq.(4.58). ■

4.9.6 Lemma 4.8

According Theorem 4.8 in Albert [5], it holds for a self-adjoint operator Z that

$$(I + \alpha^{-1}Z)^{-1} = (I - ZZ^\dagger) + \alpha Z^\dagger(I + \alpha Z^\dagger)^{-1}, \tag{4.166}$$

and it generally holds that

$$(I + \alpha Z^\dagger)^{-1} = I - \alpha Z^\dagger(I + \alpha Z^\dagger)^{-1}. \tag{4.167}$$

If Eq.(4.167) is repeatedly applied to Eq.(4.166), we have

$$\begin{aligned}(I+\alpha^{-1}Z)^{-1} &= (I-ZZ^\dagger)+\alpha Z^\dagger\left[I-\alpha Z^\dagger(I+\alpha Z^\dagger)^{-1}\right]\\ &= (I-ZZ^\dagger)+\alpha Z^\dagger-(\alpha Z^\dagger)^2(I+\alpha Z^\dagger)^{-1}\\ &= (I-ZZ^\dagger)+\alpha Z^\dagger-(\alpha Z^\dagger)^2\left[I-\alpha Z^\dagger(I+\alpha Z^\dagger)^{-1}\right]\\ &= (I-ZZ^\dagger)+\alpha Z^\dagger-(\alpha Z^\dagger)^2+(\alpha Z^\dagger)^3(I+\alpha Z^\dagger)^{-1}\\ &\vdots\\ &= (I-ZZ^\dagger)-\sum_{j=1}^{n}(-\alpha Z^\dagger)^j-(-\alpha Z^\dagger)^{n+1}(I+\alpha Z^\dagger)^{-1}, \end{aligned} \tag{4.168}$$

where n is an arbitrary fixed positive integer. Then it follows from Eqs.(4.61) and (4.168) with $Z = A^*A$ that

$$\begin{aligned}X_\alpha &= (A^*A+\alpha I_H)^{-1}A^*\\ &= (\alpha(\alpha^{-1}A^*A+I_H))^{-1}A^*\\ &= \alpha^{-1}(I_H+\alpha^{-1}A^*A)^{-1}A^*\\ &= \alpha^{-1}\Bigg(-\sum_{j=1}^{n}(-\alpha)^j(A^*A)^{-j}\\ &\qquad -(-\alpha)^{n+1}(A^*A)^{-(n+1)}(I_H+\alpha(A^*A)^{-1})^{-1}\Bigg)A^*\\ &= \sum_{j=1}^{n}(-\alpha)^{j-1}(A^*A)^{-j}A^*\\ &\quad +(-\alpha)^n(A^*A)^{-(n+1)}(I_H+\alpha(A^*A)^{-1})^{-1}A^*, \end{aligned} \tag{4.169}$$

which implies Eq.(4.63). ■

4.9.7 Lemma 4.9

According to the eigenvalue decomposition, eigenvalues of $\alpha^3(A^*A)^{-3}$ are $\mathcal{O}\left((\lambda_{\max}\alpha)^3\right)$. If terms dominated by $\alpha^3(A^*A)^{-3}$ are ignored, it follows from Eq.(4.63) that

$$X_\alpha \approx (A^*A)^{-1}A^*-\alpha(A^*A)^{-2}A^*+\alpha^2(A^*A)^{-3}A^*. \tag{4.170}$$

Then it holds that

$$\begin{aligned}X_\alpha X_\alpha^* &\approx \left((A^*A)^{-1}A^*-\alpha(A^*A)^{-2}A^*+\alpha^2(A^*A)^{-3}A^*\right)\\ &\quad\times\left(A(A^*A)^{-1}-\alpha A(A^*A)^{-2}+\alpha^2A(A^*A)^{-3}\right)\\ &\approx (A^*A)^{-1}-2\alpha(A^*A)^{-2}+3\alpha^2(A^*A)^{-3}. \end{aligned} \tag{4.171}$$

It follows from Eqs.(4.170) and (4.28) that

$$\begin{aligned} X_\alpha - A^\dagger &\approx (A^*A)^{-1}A^* - \alpha(A^*A)^{-2}A^* + \alpha^2(A^*A)^{-3}A^* - (A^*A)^{-1}A^* \\ &= -\alpha(A^*A)^{-2}A^* + \alpha^2(A^*A)^{-3}A^*. \end{aligned} \tag{4.172}$$

Hence, we have

$$\begin{aligned} (X_\alpha - A^\dagger)^*(X_\alpha - A^\dagger) &\approx \left(-\alpha A(A^*A)^{-2} + \alpha^2 A(A^*A)^{-3}\right) \\ &\quad \times \left(-\alpha(A^*A)^{-2}A^* + \alpha^2(A^*A)^{-3}A^*\right) \\ &\approx \alpha^2 A(A^*A)^{-4}A^*. \end{aligned} \tag{4.173}$$

Then it follows from Eqs.(4.62), (4.173), and (4.171) that

$$\begin{aligned} \mathrm{SIC}[\alpha] &= \|(X_\alpha - A^\dagger)\boldsymbol{y}\|^2 - \hat{\sigma}^2\mathrm{tr}(X_\alpha - A^\dagger)(X_\alpha - A^\dagger)^* + \hat{\sigma}^2\mathrm{tr}X_\alpha X_\alpha^* \\ &= \langle (X_\alpha - A^\dagger)^*(X_\alpha - A^\dagger)\boldsymbol{y}, \boldsymbol{y}\rangle - \hat{\sigma}^2\mathrm{tr}(X_\alpha - A^\dagger)^*(X_\alpha - A^\dagger) + \hat{\sigma}^2\mathrm{tr}X_\alpha X_\alpha^* \\ &\approx \alpha^2\langle A(A^*A)^{-4}A^*\boldsymbol{y}, \boldsymbol{y}\rangle - \alpha^2\hat{\sigma}^2\mathrm{tr}A(A^*A)^{-4}A^* \\ &\quad + \hat{\sigma}^2\mathrm{tr}\left((A^*A)^{-1} - 2\alpha(A^*A)^{-2} + 3\alpha^2(A^*A)^{-3}\right) \\ &= \alpha^2\left(\|(A^*A)^{-2}A^*\boldsymbol{y}\|^2 + 2\hat{\sigma}^2\mathrm{tr}(A^*A)^{-3}\right) \\ &\quad - 2\alpha\hat{\sigma}^2\mathrm{tr}(A^*A)^{-2} + \hat{\sigma}^2\mathrm{tr}(A^*A)^{-1}, \end{aligned} \tag{4.174}$$

which concludes the proof. ■

4.9.8 Theorem 4.10

From Eq.(4.64), $\widehat{\mathrm{SIC}}$ is expressed as

$$\begin{aligned} \widehat{\mathrm{SIC}}[\alpha] &= \left(\|(A^*A)^{-2}A^*\boldsymbol{y}\|^2 + 2\hat{\sigma}^2\mathrm{tr}(A^*A)^{-3}\right) \\ &\quad \times \left(\alpha - \frac{\hat{\sigma}^2\mathrm{tr}(A^*A)^{-2}}{\|(A^*A)^{-2}A^*\boldsymbol{y}\|^2 + 2\hat{\sigma}^2\mathrm{tr}(A^*A)^{-3}}\right)^2 \\ &\quad - \frac{\left(\hat{\sigma}^2\mathrm{tr}(A^*A)^{-2}\right)^2}{\|(A^*A)^{-2}A^*\boldsymbol{y}\|^2 + 2\hat{\sigma}^2\mathrm{tr}(A^*A)^{-3}} + \hat{\sigma}^2\mathrm{tr}(A^*A)^{-1}, \end{aligned} \tag{4.175}$$

which is minimized if and only if $\alpha = \frac{\hat{\sigma}^2\mathrm{tr}(A^*A)^{-2}}{\|(A^*A)^{-2}A^*\boldsymbol{y}\|^2 + 2\hat{\sigma}^2\mathrm{tr}(A^*A)^{-3}}$. For any α, it follows from Eqs.(4.66) and (4.67) that

$$\mathrm{SIC}[\alpha_{SIC}] \leq \mathrm{SIC}[\alpha], \tag{4.176}$$

$$\widehat{\mathrm{SIC}}[\alpha_{\widehat{SIC}}] \leq \widehat{\mathrm{SIC}}[\alpha]. \tag{4.177}$$

It follows from Eq.(4.176) that

$$\mathrm{SIC}[\alpha_{\widehat{SIC}}] - \mathrm{SIC}[\alpha_{SIC}] \geq 0. \tag{4.178}$$

It follows from Eq.(4.177) that

$$\begin{aligned}\mathrm{SIC}[\alpha_{\widehat{SIC}}] - \mathrm{SIC}[\alpha_{SIC}] &= \mathrm{SIC}[\alpha_{\widehat{SIC}}] - \widehat{\mathrm{SIC}}[\alpha_{\widehat{SIC}}] + \widehat{\mathrm{SIC}}[\alpha_{\widehat{SIC}}] - \mathrm{SIC}[\alpha_{SIC}] \\ &\leq \left(\mathrm{SIC}[\alpha_{\widehat{SIC}}] - \widehat{\mathrm{SIC}}[\alpha_{\widehat{SIC}}]\right) \\ &\quad + \left(\widehat{\mathrm{SIC}}[\alpha_{SIC}] - \mathrm{SIC}[\alpha_{SIC}]\right). \end{aligned} \tag{4.179}$$

Eqs.(4.178), (4.179), and (4.65) imply Eq.(4.69). ■

4.9.9 Corollary 4.11

It follows from Eqs.(4.157) and (4.71) that

$$\begin{aligned}(A^*A)^{-1} &= A^\dagger (A^\dagger)^* = W B^\dagger (B^\dagger)^* W^* \\ &= WYW^*. \end{aligned} \tag{4.180}$$

Then it follows from Eqs.(4.68), (4.180), (4.156), (4.159), and (4.40) that

$$\begin{aligned}\alpha_{\widehat{SIC}} &= \frac{\hat{\sigma}^2 \mathrm{tr} WYW^*WYW^*}{\|WYW^*WYW^*(W^{-1})^*B^*\boldsymbol{y}\|^2 + 2\hat{\sigma}^2 \mathrm{tr} WYW^*WYW^*WYW^*} \\ &= \frac{\hat{\sigma}^2 \mathrm{tr}(UY)^2}{\|WYUYB^*\boldsymbol{y}\|^2 + 2\hat{\sigma}^2 \mathrm{tr}(UY)^3} \\ &= \frac{\hat{\sigma}^2 \mathrm{tr}(UY)^2}{\langle (UY)^3 B^*\boldsymbol{y}, YB^*\boldsymbol{y}\rangle + 2\hat{\sigma}^2 \mathrm{tr}(UY)^3}, \end{aligned} \tag{4.181}$$

which implies Eq.(4.70). ■

4.9.10 Theorem 4.12

Operating $A^\dagger$ from the right-hand side of Eq.(4.73), we have

$$A^\dagger = \frac{1}{M} A^*. \tag{4.182}$$

It follows from Eqs.(4.74), (4.182), and (4.73) that

$$X_\alpha - A^\dagger = \frac{1}{M+\alpha} A^* - \frac{1}{M} A^* = -\frac{\alpha}{M(M+\alpha)} A^*, \tag{4.183}$$

$$X_\alpha X_\alpha^* = \frac{1}{(M+\alpha)^2} A^*A = \frac{M}{(M+\alpha)^2} I_H. \tag{4.184}$$

It follows from Eqs.(4.73), (4.182), (4.30), (4.8), (4.26), and (4.31) that

$$\begin{aligned}\frac{1}{M^2}\|A^*\boldsymbol{y}\|^2 &= \frac{1}{M^2}\langle AA^*\boldsymbol{y},\boldsymbol{y}\rangle = \frac{1}{M^2}\langle A(\frac{1}{M}A^*A)A^*\boldsymbol{y},\boldsymbol{y}\rangle = \frac{1}{M^3}\|AA^*\boldsymbol{y}\|^2 \\ &= \frac{1}{M}\|AA^\dagger\boldsymbol{y}\|^2 = -\frac{1}{M}\|AA^\dagger\boldsymbol{y}-\boldsymbol{y}\|^2 + \frac{1}{M}\|\boldsymbol{y}\|^2 \\ &= -\frac{1}{M}\|A\hat{f}_u-\boldsymbol{y}\|^2 + \frac{1}{M}\|\boldsymbol{y}\|^2 = -\frac{1}{M}\sum_{m=1}^{M}\left|\hat{f}_u(\boldsymbol{x}_m)-y_m\right|^2 + \frac{1}{M}\|\boldsymbol{y}\|^2 \\ &= -\frac{1}{M}(M-\mu)\hat{\sigma}^2 + \frac{1}{M}\|\boldsymbol{y}\|^2 = -\hat{\sigma}^2 + \frac{\hat{\sigma}^2\mu}{M} + \frac{1}{M}\|\boldsymbol{y}\|^2. \end{aligned} \tag{4.185}$$

Then it follows from Eqs.(4.62), (4.183), (4.184), (4.73), and (4.185) that

$$\begin{aligned}\mathrm{SIC}[\alpha] &= \|(X_\alpha - A^\dagger)\boldsymbol{y}\|^2 - \hat{\sigma}^2\mathrm{tr}(X_\alpha - A^\dagger)(X_\alpha - A^\dagger)^* + \hat{\sigma}^2\mathrm{tr}X_\alpha X_\alpha^* \\ &= \|\frac{\alpha}{M(M+\alpha)}A^*\boldsymbol{y}\|^2 - \frac{\alpha^2\hat{\sigma}^2}{M^2(M+\alpha)^2}\mathrm{tr}A^*A + \frac{\hat{\sigma}^2 M}{(M+\alpha)^2}\mathrm{tr}I_H \\ &= \frac{1}{(M+\alpha)^2}\left(\alpha^2\frac{\|A^*\boldsymbol{y}\|^2}{M^2} - \alpha^2\frac{\hat{\sigma}^2}{M}\mathrm{tr}I_H + \hat{\sigma}^2 M\mathrm{tr}I_H\right) \\ &= \frac{1}{(M+\alpha)^2}\left(\alpha^2(-\hat{\sigma}^2 + \frac{\hat{\sigma}^2\mu}{M} + \frac{\|\boldsymbol{y}\|^2}{M}) - \alpha^2\frac{\hat{\sigma}^2\mu}{M} + \hat{\sigma}^2 M\mu\right) \\ &= \frac{\alpha^2(\frac{1}{M}\|\boldsymbol{y}\|^2 - \hat{\sigma}^2) + \hat{\sigma}^2 M\mu}{(M+\alpha)^2}. \end{aligned} \tag{4.186}$$

The derivative of SIC with respect to α is given as

$$\frac{d}{d\alpha}\mathrm{SIC}[\alpha] = \frac{2M(\alpha(\frac{1}{M}\|\boldsymbol{y}\|^2 - \hat{\sigma}^2) - \hat{\sigma}^2\mu)}{(M+\alpha)^3}. \tag{4.187}$$

$\frac{d}{d\alpha}$SIC is negative if $\alpha < \frac{\hat{\sigma}^2\mu}{\frac{1}{M}\|\boldsymbol{y}\|^2 - \hat{\sigma}^2}$, $\frac{d}{d\alpha}$SIC is positive if $\alpha > \frac{\hat{\sigma}^2\mu}{\frac{1}{M}\|\boldsymbol{y}\|^2 - \hat{\sigma}^2}$, and $\frac{d}{d\alpha}$SIC vanishes if $\alpha = \frac{\hat{\sigma}^2\mu}{\frac{1}{M}\|\boldsymbol{y}\|^2 - \hat{\sigma}^2}$. Therefore, SIC is minimized if and only if $\alpha = \frac{\hat{\sigma}^2\mu}{\frac{1}{M}\|\boldsymbol{y}\|^2 - \hat{\sigma}^2}$. ■

4.9.11 Lemma 4.15

It follows from Eqs.(4.6), (4.11), (4.26), (4.79), (4.9), (4.77), and (4.73) that

$$\begin{aligned}\mathrm{E}_{\boldsymbol{\epsilon}}\left(\frac{1}{M}\sum_{m=1}^{M}|y_m|^2 - \hat{\sigma}^2\right) &= \mathrm{E}_{\boldsymbol{\epsilon}}\left(\frac{\|\boldsymbol{y}\|^2}{M} - \hat{\sigma}^2\right) = \mathrm{E}_{\boldsymbol{\epsilon}}\left(\frac{1}{M}\|Af+\boldsymbol{\epsilon}\|^2\right) - \sigma^2 \\ &= \mathrm{E}_{\boldsymbol{\epsilon}}\left(\frac{1}{M}\|Af\|^2 + \frac{2}{M}\mathrm{Re}\langle Af,\boldsymbol{\epsilon}\rangle + \frac{1}{M}\|\boldsymbol{\epsilon}\|^2\right) - \sigma^2 \\ &= \frac{\|Af\|^2}{M} + \frac{\sigma^2}{M}\mathrm{tr}I_M - \sigma^2 = \frac{\langle A^*Af, f\rangle}{M} \\ &= \|f\|^2, \end{aligned} \tag{4.188}$$

which implies Eq.(4.80). ■

Chapter 5

Theory of active learning

5.1 Introduction

Supervised learning can be classified into two categories depending on the type of sampling. One is the case where training examples are given unilaterally from the environment. For example, in *time series prediction*, sample points are fixed to regular intervals and learners can not change the interval. The other is the case where learners can design the location of sample points by themselves, and gather corresponding sample values. For example, it is possible to design sampling locations in many scientific experiments or learning of *sensorimotor maps* of multi-joint robot arms. If we can actively design the sampling locations, then a higher level of the generalization capability is expected to be acquired. (see Figure 1.5 in page 9).

The problem of designing the sampling locations for the optimal generalization capability is called *active learning* (Cohn *et al.* [27], Fukumizu [38], Vijayakumar & Ogawa [143]). It is also referred to as *optimal experiments* (Kiefer [60], Fedorov [34], Cohn [25]) or *query construction* (Sollich [124]).

From the viewpoint of the optimality, active learning can be classified into two categories. One is the *global optimality*, where a set of all sample points is optimal (e.g. Fedorov [34], Vijayakumar & Ogawa [143], Yue & Hickernell [149]). The other is the *greedy optimality*, where the next sample point to add is optimal in each step (e.g. MacKay [70], Cohn [25], Fukumizu [38]). In this chapter, we devise both global and greedy optimal active learning methods. The global method gives exactly the optimal generalization capability for trigonometric polynomial models. In contrast, the greedy method is applicable to any finite dimensional models.

5.2 Problem formulation

Let $\mathcal{X}$ be a set of M sample points:

$$\mathcal{X} = \{\boldsymbol{x}_m\}_{m=1}^{M}. \tag{5.1}$$

The sample values $\{y_m\}_{m=1}^{M}$ are degraded by additive noise $\{\epsilon_m\}_{m=1}^{M}$:

$$y_m = f(\boldsymbol{x}_m) + \epsilon_m. \tag{5.2}$$

Let us denote the learning result function obtained with $\mathcal{X}$ by $\hat{f}_{\mathcal{X}}(\boldsymbol{x})$. We assume that the learning target function $f(\boldsymbol{x})$ and the learning result function $\hat{f}_{\mathcal{X}}(\boldsymbol{x})$ belong to a specified reproducing kernel Hilbert space H (see Section 2.3), and the generalization error of $\hat{f}_{\mathcal{X}}(\boldsymbol{x})$ is measured by

$$J_G[\mathcal{X}] = \mathrm{E}_{\boldsymbol{\epsilon}}\|\hat{f}_{\mathcal{X}} - f\|^2, \tag{5.3}$$

where $\mathrm{E}_{\boldsymbol{\epsilon}}$ denotes the expectation over the noise. The norm is typically defined as

$$\|\hat{f}_{\mathcal{X}} - f\|^2 = \int \left|\hat{f}_{\mathcal{X}}(\boldsymbol{u}) - f(\boldsymbol{u})\right|^2 w(\boldsymbol{u})d\boldsymbol{u}, \tag{5.4}$$

where the integral with respect to $\boldsymbol{u}$ means the expectation over future sample points $\boldsymbol{u}$ and $w(\boldsymbol{u})$ is some weight function, e.g., the probability density function of $\boldsymbol{u}$. Then the problem of active learning considered in this chapter is formulated as follows.

Definition 5.1 (Active learning) *Determine the best set $\hat{\mathcal{X}}$ of sample points that minimizes the generalization error J_G:*

$$\hat{\mathcal{X}} = \operatorname*{argmin}_{\mathcal{X}} J_G[\mathcal{X}]. \tag{5.5}$$

5.3 Batch active learning

In this section, we propose a global optimal active learning method, where sample points are determined in a batch manner. For this reason, it is called *batch active learning*. We give a necessary and sufficient condition of sample points for providing the globally optimal generalization capability.

5.3.1 Setting

First, the setting is described.

1. The function space H to which the learning target function $f(\boldsymbol{x})$ belongs is finite dimensional:

$$\dim H < \infty. \tag{5.6}$$

2. The reproducing kernel $K(\boldsymbol{x}, \boldsymbol{x}')$ of H satisfies

$$K(\boldsymbol{x}, \boldsymbol{x}) = r \quad \text{for any } \boldsymbol{x} \in \mathcal{D}, \tag{5.7}$$

where r is a non-negative constant.

3. The learning result function is obtained by LMS learning for the model H (see Section 3.2.1). In this case, the learning result function $\hat{f}_{\mathcal{X}}(\boldsymbol{x})$ obtained with the sample points $\mathcal{X}$ is given as

$$\hat{f}_{\mathcal{X}} = A_{\mathcal{X}}^{\dagger} \boldsymbol{y}, \tag{5.8}$$

where $A_{\mathcal{X}}$ is defined as

$$A_{\mathcal{X}} = \sum_{m=1}^{M} \left(\boldsymbol{e}_m \otimes \overline{K(\cdot, \boldsymbol{x}_m)} \right). \tag{5.9}$$

Here, $(\cdot \otimes \bar{\cdot})$ denotes the Neumann-Schatten product (see Section 2.2.4) and $\boldsymbol{e}_m$ is the m-th vector of the so-called standard basis in $\mathbf{C}^M$. The vector $\boldsymbol{y}$ is defined as

$$\boldsymbol{y} = (y_1, y_2, \ldots, y_M)^{\top}, \tag{5.10}$$

where $\top$ denotes the transpose of a vector. It holds for any function f in H that

$$A_{\mathcal{X}} f = \boldsymbol{z}, \tag{5.11}$$

where $\boldsymbol{z}$ is the M-dimensional vector is defined as

$$\boldsymbol{z} = (f(\boldsymbol{x}_1), f(\boldsymbol{x}_2), \ldots, f(\boldsymbol{x}_M))^{\top}. \tag{5.12}$$

This can be verified from the property of the reproducing kernel (see Section 2.3):

$$\langle f(\cdot), K(\cdot, \boldsymbol{x}') \rangle = f(\boldsymbol{x}'). \tag{5.13}$$

4. The functions $\{K(\boldsymbol{x}, \boldsymbol{x}_m)\}_{m=1}^M$ span the whole space H:

$$\mathcal{L}\left(\{K(\boldsymbol{x}, \boldsymbol{x}_m)\}_{m=1}^M\right) = H, \tag{5.14}$$

where $\mathcal{L}\left(\{\varphi_p(\boldsymbol{x})\}_p\right)$ denotes the linear manifold spanned by $\{\varphi_p(\boldsymbol{x})\}_p$. Eq.(5.14) is equivalently expressed as

$$\mathcal{R}(A_{\mathcal{X}}^*) = H, \tag{5.15}$$

where $A_{\mathcal{X}}^*$ denotes the adjoint operator of $A_{\mathcal{X}}$. Note that Eq.(5.14) holds only if the number M of training examples is larger than or equal to the dimension of H:

$$M \geq \dim H. \tag{5.16}$$

5. For any sample points $\{\boldsymbol{x}_m\}_{m=1}^M$, the mean noise is zero:

$$\mathrm{E}_{\boldsymbol{\epsilon}}\boldsymbol{\epsilon} = 0, \tag{5.17}$$

where

$$\boldsymbol{\epsilon} = (\epsilon_1, \epsilon_2, \ldots, \epsilon_M)^\top. \tag{5.18}$$

6. For any sample points $\{\boldsymbol{x}_m\}_{m=1}^M$, the noise covariance matrix Q is in the form

$$Q = \mathrm{E}_{\boldsymbol{\epsilon}}\left(\boldsymbol{\epsilon} \otimes \overline{\boldsymbol{\epsilon}}\right) = \sigma^2 I_M \tag{5.19}$$

with $\sigma^2 > 0$, where I_M is the M-dimensional identity matrix. σ^2 does not have to be known.

5.3.2 Necessary and sufficient condition for optimal generalization

Under the above setting, we shall derive the condition for the optimal generalization capability.

It is known that the generalization error of $\hat{f}_{\mathcal{X}}(\boldsymbol{x})$ can be decomposed into the bias and variance (see e.g. Takemura [132], Geman *et al.* [40], Efron & Tibshirani [33]):

$$J_G[\mathcal{X}] = \|\mathrm{E}_{\boldsymbol{\epsilon}}\hat{f}_{\mathcal{X}} - f\|^2 + \mathrm{E}_{\boldsymbol{\epsilon}}\|\hat{f}_{\mathcal{X}} - \mathrm{E}_{\boldsymbol{\epsilon}}\hat{f}_{\mathcal{X}}\|^2. \tag{5.20}$$

It follows from Eqs.(5.2), (5.10), (5.12), and (5.18) that

$$\boldsymbol{y} = \boldsymbol{z} + \boldsymbol{\epsilon}. \tag{5.21}$$

Then it follows from Eqs.(5.8), (5.21), (5.17), (5.11), and (5.15) that the bias of $\hat{f}_{\mathcal{X}}(\boldsymbol{x})$ vanishes:

$$\begin{aligned}\|\mathrm{E}_{\boldsymbol{\epsilon}}\hat{f}_{\mathcal{X}} - f\|^2 &= \|\mathrm{E}_{\boldsymbol{\epsilon}}A_{\mathcal{X}}^{\dagger}\boldsymbol{y} - f\|^2 = \|\mathrm{E}_{\boldsymbol{\epsilon}}A_{\mathcal{X}}^{\dagger}(\boldsymbol{z}+\boldsymbol{\epsilon}) - f\|^2 \\ &= \|A_{\mathcal{X}}^{\dagger}\boldsymbol{z} - f\|^2 = \|A_{\mathcal{X}}^{\dagger}A_{\mathcal{X}}f - f\|^2 \\ &= \|P_{\mathcal{R}(A_{\mathcal{X}}^*)}f - f\|^2 = \|I_H f - f\|^2 \\ &= 0, \end{aligned} \tag{5.22}$$

where $P_{\mathcal{R}(A_{\mathcal{X}}^*)}$ denotes the orthogonal projection operator onto the range of $A_{\mathcal{X}}^*$ and I_H denotes the identity operator on H. Therefore, it follows from Eqs.(5.20), (5.22), (5.8), (5.21), (5.17), and (5.19) that the generalization error of $\hat{f}_{\mathcal{X}}(\boldsymbol{x})$ is reduced to as

$$\begin{aligned}J_G[\mathcal{X}] &= \mathrm{E}_{\boldsymbol{\epsilon}}\|\hat{f}_{\mathcal{X}} - \mathrm{E}_{\boldsymbol{\epsilon}}\hat{f}_{\mathcal{X}}\|^2 = \mathrm{E}_{\boldsymbol{\epsilon}}\|A_{\mathcal{X}}^{\dagger}\boldsymbol{y} - \mathrm{E}_{\boldsymbol{\epsilon}}A_{\mathcal{X}}^{\dagger}\boldsymbol{y}\|^2 \\ &= \mathrm{E}_{\boldsymbol{\epsilon}}\|A_{\mathcal{X}}^{\dagger}(\boldsymbol{z}+\boldsymbol{\epsilon}) - \mathrm{E}_{\boldsymbol{\epsilon}}A_{\mathcal{X}}^{\dagger}(\boldsymbol{z}+\boldsymbol{\epsilon})\|^2 = \mathrm{E}_{\boldsymbol{\epsilon}}\|A_{\mathcal{X}}^{\dagger}(\boldsymbol{z}+\boldsymbol{\epsilon}) - A_{\mathcal{X}}^{\dagger}\boldsymbol{z}\|^2 \\ &= \mathrm{E}_{\boldsymbol{\epsilon}}\|A_{\mathcal{X}}^{\dagger}\boldsymbol{\epsilon}\|^2 = \mathrm{tr}\left(A_{\mathcal{X}}^{\dagger}\mathrm{E}_{\boldsymbol{\epsilon}}\left(\boldsymbol{\epsilon}\otimes\overline{\boldsymbol{\epsilon}}\right)\left(A_{\mathcal{X}}^{\dagger}\right)^*\right) \\ &= \sigma^2\mathrm{tr}A_{\mathcal{X}}^{\dagger}(A_{\mathcal{X}}^{\dagger})^*, \end{aligned} \tag{5.23}$$

where 'tr' denotes the trace of an operator. Consequently, the problem of active learning considered in this section becomes the problem of finding a set $\mathcal{X}$ of sample points that minimizes Eq.(5.23) under the constraint of Eq.(5.14). Then we have the following theorem.

Theorem 5.2 *The generalization error J_G defined by Eq.(5.3) is minimized with respect to the operator $A_{\mathcal{X}}$ under the constraint of Eq.(5.14) if and only if*

$$\frac{\mu}{rM}A_{\mathcal{X}}^*A_{\mathcal{X}} = I_H, \tag{5.24}$$

where μ is the dimension of H, r is given by Eq.(5.7), and I_H denotes the identity operator on H. In this case, the minimum value of J_G is given as

$$\frac{\sigma^2\mu^2}{rM}. \tag{5.25}$$

A proof of Theorem 5.2 is provided in Section 5.8.1.

Eq.(5.24) is equivalent to that a set $\{\sqrt{\frac{\mu}{rM}}K(\cdot,\boldsymbol{x}_m)\}_{m=1}^M$ forms a *pseudo orthonormal basis* (PONB) (Ogawa & Iijima [92], Ogawa [91]) in H, which is an extension of orthonormal bases to linearly dependent over-complete systems. The rigorous definition and properties of PONBs are described in Section 5.7. By using the properties of PONBs, we have the following lemma.

Lemma 5.3 *When the operator $A_{\mathcal{X}}$ satisfies Condition (5.24), it holds that*

$$\|A_{\mathcal{X}} f\| = \sqrt{\frac{rM}{\mu}} \|f\| \quad \text{for any } f \in H, \tag{5.26}$$

$$\|A_{\mathcal{X}}^{\dagger} \boldsymbol{v}\| = \begin{cases} \sqrt{\frac{\mu}{rM}} \|\boldsymbol{v}\| & \text{for any } \boldsymbol{v} \in \mathcal{R}(A_{\mathcal{X}}), \\ 0 & \text{for any } \boldsymbol{v} \in \mathcal{R}(A_{\mathcal{X}})^{\perp}, \end{cases} \tag{5.27}$$

where $\mathcal{R}(A_{\mathcal{X}})$ is the range of $A_{\mathcal{X}}$ and $\mathcal{R}(A_{\mathcal{X}})^{\perp}$ denotes the orthogonal complement of $\mathcal{R}(A_{\mathcal{X}})$.

A proof of Lemma 5.3 is given in Section 5.8.2.

Eqs.(5.26) and (5.27) imply that $\sqrt{\frac{\mu}{rM}} A_{\mathcal{X}}$ becomes an *isometry* and $\sqrt{\frac{rM}{\mu}} A_{\mathcal{X}}^{\dagger}$ becomes a *partial isometry* with the initial space $\mathcal{R}(A_{\mathcal{X}})$, respectively (see Section 2.2.3).

Lemma 5.3 gives interpretation of Theorem 5.2. Let us decompose the noise $\boldsymbol{\epsilon}$ into $\tilde{\boldsymbol{\epsilon}} \in \mathcal{R}(A_{\mathcal{X}})$ and $\tilde{\boldsymbol{\epsilon}}^{\perp} \in \mathcal{R}(A_{\mathcal{X}})^{\perp}$:

$$\boldsymbol{\epsilon} = \tilde{\boldsymbol{\epsilon}} + \tilde{\boldsymbol{\epsilon}}^{\perp}. \tag{5.28}$$

Then it follows from Eqs.(5.21), (5.11), and (5.28) that the sample value vector $\boldsymbol{y}$ is rewritten as

$$\boldsymbol{y} = A_{\mathcal{X}} f + \tilde{\boldsymbol{\epsilon}} + \tilde{\boldsymbol{\epsilon}}^{\perp}. \tag{5.29}$$

Because of Eq.(5.15), it holds for any f in H that

$$A_{\mathcal{X}}^{\dagger} A_{\mathcal{X}} f = P_{\mathcal{R}(A_{\mathcal{X}}^{*})} f = I_H f = f, \tag{5.30}$$

which implies that the signal component $A_{\mathcal{X}} f$ in Eq.(5.29) is transformed to the original function f by $A_{\mathcal{X}}^{\dagger}$. From Eq.(5.27), $A_{\mathcal{X}}^{\dagger}$ suppresses the magnitude of the noise $\tilde{\boldsymbol{\epsilon}}$ in $\mathcal{R}(A_{\mathcal{X}})$ by $\sqrt{\frac{\mu}{rM}}$ and completely removes the noise $\tilde{\boldsymbol{\epsilon}}^{\perp}$ in $\mathcal{R}(A_{\mathcal{X}})^{\perp}$:

$$\|A_{\mathcal{X}}^{\dagger} \tilde{\boldsymbol{\epsilon}}\| = \sqrt{\frac{\mu}{rM}} \|\tilde{\boldsymbol{\epsilon}}\|, \tag{5.31}$$

$$A_{\mathcal{X}}^{\dagger} \tilde{\boldsymbol{\epsilon}}^{\perp} = 0. \tag{5.32}$$

The above analysis is summarized in Figure 5.1.

In general, it is difficult to suppress the effect of the noise $\tilde{\boldsymbol{\epsilon}}$ in $\mathcal{R}(A_{\mathcal{X}})$ since it can not be distinguished from the signal component $A_{\mathcal{X}} f$. However, the above analysis suggests that the effect of the noise $\tilde{\boldsymbol{\epsilon}}$ is minimized if the mean magnification of $A_{\mathcal{X}}^{\dagger}$ is minimized. Since minimizing the mean magnification of $A_{\mathcal{X}}^{\dagger}$ is equivalent to maximizing the mean magnification of $A_{\mathcal{X}}$, the effect of the noise $\tilde{\boldsymbol{\epsilon}}$ is minimized if the norm of $A_{\mathcal{X}} f$ is maximized in the average sense. This principle well agrees with our intuition that the sampling with the highest signal-to-noise ratio in the sample value vector $\boldsymbol{y}$ provides the optimal generalization capability.

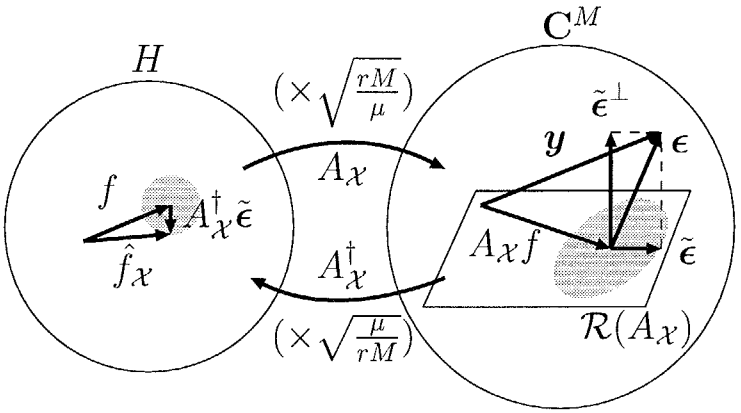


Figure 5.1: Mechanism of achieving optimal generalization capability by Theorem 5.2. If $\frac{\mu}{rM}A_{\mathcal{X}}^{*}A_{\mathcal{X}} = I_H$, then $A_{\mathcal{X}}^{\dagger}A_{\mathcal{X}}f = f$, $\|A_{\mathcal{X}}^{\dagger}\tilde{\epsilon}\| = \sqrt{\frac{\mu}{rM}}\|\tilde{\epsilon}\|$, and $A_{\mathcal{X}}^{\dagger}\tilde{\epsilon}^{\perp} = 0$.

5.3.3 Calculation of least mean squares learning functions

Now we discuss the calculation method of the LMS learning function $\hat{f}_{\mathcal{X}}(\boldsymbol{x})$ given by Eq.(5.8).

Let $\{\varphi_p(\boldsymbol{x})\}_{p=1}^{\mu}$ be an orthonormal basis in H. Then the following proposition holds.

Proposition 5.4 *(Efron & Tibshirani [33]) For general sample points, the LMS learning function $\hat{f}_{\mathcal{X}}(\boldsymbol{x})$ can be calculated as*

$$\hat{f}_{\mathcal{X}}(\boldsymbol{x}) = \sum_{p=1}^{\mu}\left(\sum_{p'=1}^{\mu}[C_{\mathcal{X}}^{-1}]_{p,p'}\sum_{m=1}^{M}\overline{\varphi_{p'}(\boldsymbol{x}_m)}y_m\right)\varphi_p(\boldsymbol{x}), \tag{5.33}$$

where $[\,\cdot\,]_{p,p'}$ denotes the (p,p')-th element of a matrix and $\overline{\cdot}$ denotes the complex conjugate of a scalar. $C_{\mathcal{X}}$ is the μ-dimensional matrix defined as

$$[C_{\mathcal{X}}]_{p,p'} = \sum_{m=1}^{M}\overline{\varphi_p(\boldsymbol{x}_m)}\varphi_{p'}(\boldsymbol{x}_m). \tag{5.34}$$

Note that Eq.(5.33) is equivalently expressed as

$$\hat{f}_{\mathcal{X}}(\boldsymbol{x}) = \sum_{p=1}^{\mu}[B_{\mathcal{X}}^{\dagger}\boldsymbol{y}]_p\varphi_p(\boldsymbol{x}), \tag{5.35}$$

where $[\,\cdot\,]_p$ denotes the p-th element of a vector, and $B_{\mathcal{X}}$ is the *design matrix* (see e.g. Efron & Tibshirani [33]), i.e., the $M \times \mu$ matrix with the (m,p)-th element being $\varphi_p(\boldsymbol{x}_m)$:

$$[B_{\mathcal{X}}]_{m,p} = \varphi_p(\boldsymbol{x}_m). \tag{5.36}$$

When the sample points satisfy Condition (5.24), the following theorem holds.

Table 5.1: Computational complexity and memory required for calculating LMS learning function $\hat{f}_{\mathcal{X}}(\boldsymbol{x})$. M is the number of training examples and μ is the dimension of H. In Corollary 5.10, H is a trigonometric polynomial space and $M = k\mu$ where k is a positive integer.

Sample Points	Calculation Method	Computational Complexity	Memory
General	Proposition 5.4	$\mathcal{O}(\mu^2(M+\mu))$	$\mathcal{O}(M+\mu^2)$
Condition (5.24)	Theorem 5.5	$\mathcal{O}(\mu M)$	$\mathcal{O}(M+\mu)$
Theorem 5.9 with Eq.(5.53)	Corollary 5.10	$\mathcal{O}(\mu^2)$	$\mathcal{O}(\mu)$

Theorem 5.5 *When Condition (5.24) holds, the LMS learning function* $\hat{f}_{\mathcal{X}}(\boldsymbol{x})$ *can be calculated as*

$$\hat{f}_{\mathcal{X}}(\boldsymbol{x}) = \sum_{p=1}^{\mu} \left(\frac{\mu}{rM} \sum_{m=1}^{M} \overline{\varphi_p(\boldsymbol{x}_m)} y_m \right) \varphi_p(\boldsymbol{x}). \tag{5.37}$$

A proof of Theorem 5.5 is given in Section 5.8.3.

Similar to Eq.(5.35), Eq.(5.37) is equivalently expressed as

$$\hat{f}_{\mathcal{X}}(\boldsymbol{x}) = \frac{\mu}{rM} \sum_{p=1}^{\mu} [B_{\mathcal{X}}^* \boldsymbol{y}]_p \varphi_p(\boldsymbol{x}). \tag{5.38}$$

Let us measure the computational complexity by the number of scalar multiplications. For general sample points, the computational complexity and memory required for calculating $\hat{f}_{\mathcal{X}}(\boldsymbol{x})$ by Eq.(5.33) are $\mathcal{O}(\mu^2(M+\mu))$ and $\mathcal{O}(M+\mu^2)$, respectively. In contrast, Theorem 5.5 states that if sample points satisfy Condition (5.24), then the computational complexity and memory can be reduced to $\mathcal{O}(\mu M)$ and $\mathcal{O}(M+\mu)$, respectively. This shows that Theorems 5.2 and 5.5 do not only provide the optimal generalization capability but also reduce the computational complexity and memory. These results are summarized in Table 5.1.

5.3.4 Optimal design of sample points in trigonometric polynomial space

In Section 5.3.2, we gave the optimality condition of sample points for a finite dimensional reproducing kernel Hilbert space such that Eq.(5.7) holds. In this section, we use the *trigonometric polynomial space* (see Section 3.3.1) as an example of such a Hilbert space, and give design methods of optimal sample points.

When H is a trigonometric polynomial space, the generalization error J_G of $\hat{f}_{\mathcal{X}}(\boldsymbol{x})$ defined by Eq.(5.3) is expressed as

$$J_G = \mathrm{E}_{\epsilon} \frac{1}{(2\pi)^L} \int \left| \hat{f}_{\mathcal{X}}(\boldsymbol{u}) - f(\boldsymbol{u}) \right|^2 d\boldsymbol{u}, \tag{5.39}$$

where L is the dimension of the input vector $\boldsymbol{x}$. This is equivalent to the typical norm given by Eq.(5.4) with $w(\boldsymbol{u}) = \frac{1}{(2\pi)^L}$. In the case of the trigonometric polynomial space, the constant r in Eq.(5.7) is given as

$$r = \mu, \tag{5.40}$$

where μ is the dimension of H. Therefore, Eq.(5.25) is reduce to

$$\frac{\sigma^2 \mu}{M}, \tag{5.41}$$

which is equivalent to the asymptotic generalization error by passive learning (Fukumizu & Watanabe [39]). This means that Eq.(5.41) can be attained with a finite number of training examples if Eq.(5.24) holds.

For the trigonometric polynomial space, we shall give design methods of sample points $\{\boldsymbol{x}_m\}_{m=1}^M$ that satisfy the optimality condition given by (5.24). For simplicity, we shall start from the case when the dimension L of the input vector $\boldsymbol{x}$ is 1, i.e., H is a trigonometric polynomial space of order N.

Corollary 5.6 *Let $M \geq \mu$ $(= 2N + 1)$ and c be an arbitrary constant such that $-\pi \leq c \leq -\pi + \frac{2\pi}{M}$. If a set $\{x_m\}_{m=1}^M$ of M sample points is fixed to*

$$x_m = c + \frac{2\pi}{M}(m-1), \tag{5.42}$$

then Condition (5.24) holds.

Corollary 5.7 *Let $M = k\mu$ where k is a positive integer. For $t = 1, 2, \ldots, k$, let c_t be an arbitrary constant such that $-\pi \leq c_t \leq -\pi + \frac{2\pi}{\mu}$. If a set*

$$\{x_m \mid m = (t-1)\mu + p, \;\; t = 1, 2, \ldots, k, \;\; p = 1, 2, \ldots, \mu\} \tag{5.43}$$

of M sample points is fixed to

$$x_m = c_t + \frac{2\pi}{\mu}(p-1), \tag{5.44}$$

then Condition (5.24) holds.

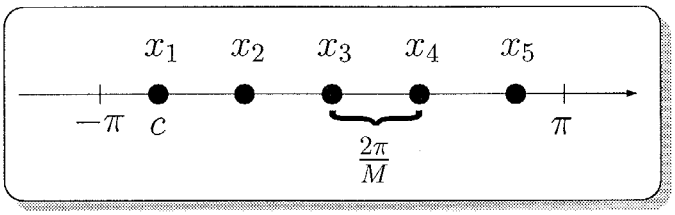


Figure 5.2: Optimal sample points for H being a trigonometric polynomial space of order 1 (Corollary 5.6). The number M of training examples is 5.

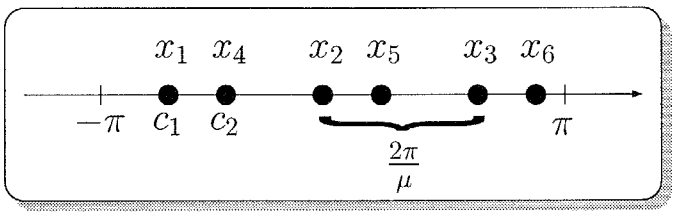


Figure 5.3: Optimal sample points for H being a trigonometric polynomial space of order 1 (Corollary 5.7). The number M of training examples is $M = k \times \mu = 2 \times 3 = 6$.

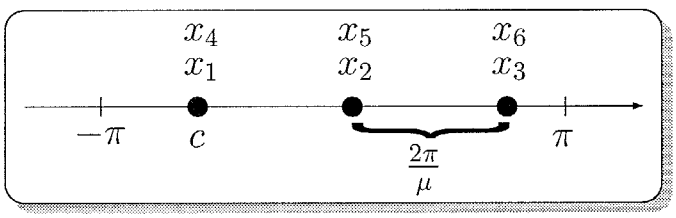


Figure 5.4: Optimal sample points for H being a trigonometric polynomial space of order 1 (Corollary 5.7 with Eq.(5.45)). The number M of training examples is $M = k \times \mu = 2 \times 3 = 6$.

Proofs of the above corollaries are omitted since we will prove the multi-dimensional case later.

Corollary 5.6 means that M sample points are fixed to regular intervals in the domain (see Figure 5.2). In contrast, Corollary 5.7 means that for each t, μ sample points are fixed to regular intervals in the domain (see Figure 5.3). Especially when

$$c_1 = c_2 = \cdots = c_k = c, \tag{5.45}$$

μ sample points are fixed to regular intervals in the domain and sample values are gathered k times at each point (see Figure 5.4). Note that the design of sample points shown in Corollary 5.6 is also *D-optimal* (see Section 5.5.2).

Now we shall give optimal design methods of sample points for multi-dimensional cases, i.e., H is a trigonometric polynomial space of order $(N_1, N_2, \ldots, N_L)$.

Theorem 5.8 *For $l = 1, 2, \ldots, L$, let M_l be a positive integer such that $M_l \geq 2N_l + 1$ and c_l be an arbitrary constant such that $-\pi \leq c_l \leq -\pi + \frac{2\pi}{M_l}$. Let the number M of training examples be*

$$M = \prod_{l=1}^{L} M_l. \tag{5.46}$$

If a set

$$\left\{ \boldsymbol{x}_m \;\middle|\; m = \sum_{l=2}^{L} \left((m_l - 1) \prod_{l'=1}^{l-1} M_{l'} \right) + m_1, \right.$$
$$\left. m_l = 1, 2, \ldots, M_l \text{ for } l = 1, 2, \ldots, L \right\} \tag{5.47}$$

of M sample points is fixed to

$$\boldsymbol{x}_m = (\xi_m^{(1)}, \xi_m^{(2)}, \ldots, \xi_m^{(L)})^\top, \tag{5.48}$$

where

$$\xi_m^{(l)} = c_l + \frac{2\pi}{M_l}(m_l - 1) \text{ for } l = 1, 2, \ldots, L, \tag{5.49}$$

then Condition (5.24) holds.

Theorem 5.9 *Let $M = k\mu$ where k is a positive integer and μ $(= \prod_{l=1}^{L}(2N_l + 1))$ is the dimension of H. For $t = 1, 2, \ldots, k$ and $l = 1, 2, \ldots, L$, let $c_{t,l}$ be an arbitrary constant such that $-\pi \leq c_{t,l} \leq -\pi + \frac{2\pi}{2N_l+1}$. If a set*

$$\left\{ \boldsymbol{x}_m \;\middle|\; m = (t-1)\mu + \sum_{l=2}^{L} \left((n_l - 1) \prod_{l'=1}^{l-1} (2N_{l'} + 1) \right) + n_1, \right.$$
$$\left. t = 1, 2, \ldots, k, \;\; n_l = 1, 2, \ldots, 2N_l + 1 \text{ for } l = 1, 2, \ldots, L \right\} \tag{5.50}$$

of M sample points is fixed to

$$\boldsymbol{x}_m = (\xi_m^{(1)}, \xi_m^{(2)}, \ldots, \xi_m^{(L)})^\top, \tag{5.51}$$

where

$$\xi_m^{(l)} = c_{t,l} + \frac{2\pi}{2N_l + 1}(n_l - 1) \text{ for } l = 1, 2, \ldots, L, \tag{5.52}$$

then Condition (5.24) holds.

Proofs of Theorems 5.8 and 5.9 are provided in Sections 5.8.4 and 5.8.5, respectively.

Examples of sample points for multi-dimensional cases are illustrated in Figures 5.5, 5.6, and 5.7. Note that in Figure 5.7, the constant $c_{t,l}$ is assigned to

$$c_{1,l} = c_{2,l} = \cdots = c_{k,l} = c_l \quad \text{for } l = 1, 2, \ldots, L. \tag{5.53}$$

As shown in Theorem 5.5, the LMS learning function can be efficiently calculated if Condition (5.24) holds. In the case of the trigonometric polynomial space, the efficiency can be further improved. Let a set $\{\varphi_p(\boldsymbol{x})\}_{p=1}^{\mu}$ be an orthonormal basis in the trigonometric polynomial space H. For example, it is given by

$$\left\{ \prod_{l=1}^{L} \exp(in_l\xi^{(l)}) \;\middle|\; n_l = -N_l, -N_l+1, \ldots, N_l \right. \\ \left. \text{for } l = 1, 2, \ldots, L \right\}. \tag{5.54}$$

Then the following corollary holds.

Corollary 5.10 *When sample points are designed following Theorem 5.9 with Eq.(5.53), the LMS learning function $\hat{f}_{\mathcal{X}}(\boldsymbol{x})$ can be calculated as*

$$\hat{f}_{\mathcal{X}}(\boldsymbol{x}) = \sum_{p=1}^{\mu} \left(\frac{1}{\mu} \sum_{p'=1}^{\mu} \overline{\varphi_p(\boldsymbol{x}_{p'})} \tilde{y}_{p'} \right) \varphi_p(\boldsymbol{x}), \tag{5.55}$$

where $\tilde{y}_{p'}$ is the mean sample value at $\boldsymbol{x}_{p'}$:

$$\tilde{y}_{p'} = \frac{1}{k} \sum_{t=1}^{k} y_{p'+(t-1)\mu}. \tag{5.56}$$

Corollary 5.10 is clear from Theorem 5.5 with Eq.(5.40). Therefore, the proof is omitted.

If sample points are designed following Theorem 5.9 with Eq.(5.53) and the LMS learning function $\hat{f}_{\mathcal{X}}(\boldsymbol{x})$ is calculated by Corollary 5.10, then the computational complexity and memory can be further reduced to $\mathcal{O}(\mu^2)$ and $\mathcal{O}(\mu)$, respectively (Table 5.1 in page 98). This is extremely efficient since the dimension μ of H does not depend on the number M of training examples.

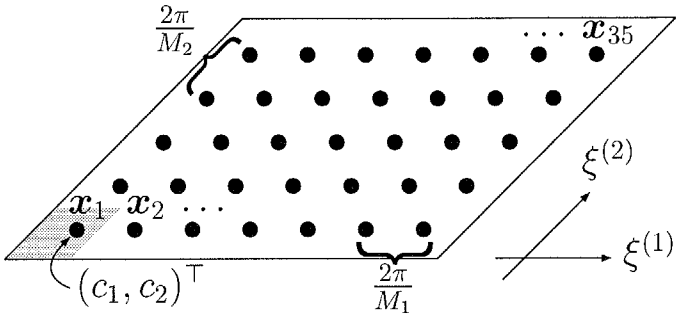


Figure 5.5: Optimal sample points for H being a trigonometric polynomial space of order $(2, 1)$ (Theorem 5.8). The number M of training examples is $M = M_1 \times M_2 = 7 \times 5 = 35$.

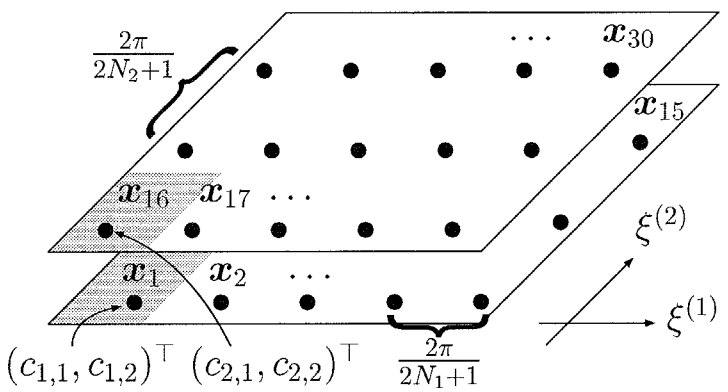


Figure 5.6: Optimal sample points for H being a trigonometric polynomial space of order $(2, 1)$ (Theorem 5.9). The number M of training examples is $M = k \times \mu = 2 \times 15 = 30$.

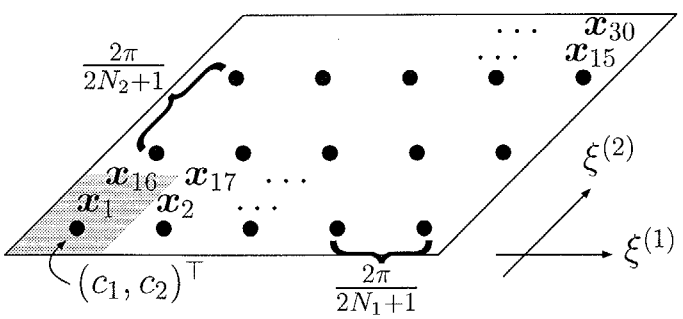


Figure 5.7: Optimal sample points for H being a trigonometric polynomial space of order $(2, 1)$ (Theorem 5.9 with Eq.(5.53)). The number M of training examples is $M = k \times \mu = 2 \times 15 = 30$.

5.4 Incremental active learning

In the previous section, we gave a global optimal active learning method, where sample points are determined in a batch manner. Although the result was very strong, the range of application was limited to models that satisfy Eq.(5.7), e.g., the trigonometric polynomial space. In this section, we propose a greedy optimal active learning method that is applicable to any finite dimensional models. In the greedy optimal method, sample points are incrementally determined. For this reason, it is called *incremental active learning*.

5.4.1 Setting

First, the setting is described.

1. The function space H to which the learning target function $f(\boldsymbol{x})$ belongs is finite dimensional:
$$\dim H < \infty. \tag{5.57}$$

2. The norm in H is computable. For example, when the norm is expressed as Eq.(5.4), the covariance operator V of the weight function $w(\boldsymbol{u})$ is assumed to be known:
$$V = \int \left(K(\cdot, \boldsymbol{u}) \otimes \overline{K(\cdot, \boldsymbol{u})}\right) w(\boldsymbol{u}) d\boldsymbol{u}, \tag{5.58}$$
where $K(\cdot, \cdot)$ is the reproducing kernel of H (see Section 2.3).

3. The number M of training examples is larger than or equal to the dimension of the function space H:
$$M \geq \dim H. \tag{5.59}$$

4. The learning result function is obtained by LMS learning for the model H (see Section 3.2.1). Let $\hat{f}_m(\boldsymbol{x})$ be the LMS learning function obtained with m training examples $\{(\boldsymbol{x}_j, y_j)\}_{j=1}^m$. Then $\hat{f}_m(\boldsymbol{x})$ is given as
$$\hat{f}_m = A_m^\dagger \boldsymbol{y}^{(m)}, \tag{5.60}$$
where A_m and $\boldsymbol{y}^{(m)}$ are defined as
$$A_m = \sum_{j=1}^m \left(\boldsymbol{e}_j^{(m)} \otimes \overline{K(\cdot, \boldsymbol{x}_j)}\right), \tag{5.61}$$
$$\boldsymbol{y}^{(m)} = (y_1, y_2, \ldots, y_m)^\top. \tag{5.62}$$

Here, $(\cdot \otimes \bar{\cdot})$ denotes the Neumann-Schatten product and $\boldsymbol{e}_j^{(m)}$ is the j-th vector of the so-called standard basis in $\mathbf{C}^m$. Note that $\boldsymbol{y}^{(m)}$ is expressed as

$$\boldsymbol{y}^{(m)} = A_m f + \boldsymbol{\epsilon}^{(m)}, \tag{5.63}$$

where $\boldsymbol{\epsilon}^{(m)}$ is the m-dimensional vector defined as

$$\boldsymbol{\epsilon}^{(m)} = (\epsilon_1, \epsilon_2, \ldots, \epsilon_m)^\top. \tag{5.64}$$

5. For any sample points $\{\boldsymbol{x}_m\}_{m=1}^M$, the mean noise is zero:

$$\mathrm{E}_{\boldsymbol{\epsilon}} \boldsymbol{\epsilon}^{(M)} = 0. \tag{5.65}$$

6. For any sample points $\{\boldsymbol{x}_m\}_{m=1}^M$, the noise covariance matrix Q is in the form

$$Q = \mathrm{E}_{\boldsymbol{\epsilon}} \left(\boldsymbol{\epsilon}^{(M)} \otimes \overline{\boldsymbol{\epsilon}^{(M)}} \right) = \sigma^2 I_M \tag{5.66}$$

with $\sigma^2 > 0$. σ^2 does not have to be known.

5.4.2 Incremental least mean squares learning

In incremental active learning, learning result functions are updated every time a new training example is added. This type of successive learning method is called *incremental learning* (Vijayakumar & Ogawa [142], Sugiyama & Ogawa [129][130]). In contrast, learning with all training examples is called *batch learning*. Here, we give an incremental learning method for LMS learning following Sugiyama and Ogawa [129][130].

Let us consider the case where a new training example $(\boldsymbol{x}_{m+1}, y_{m+1})$ is added after the LMS learning function $\hat{f}_m$ has been obtained. It follows from Eq.(5.60) that the LMS learning function $\hat{f}_{m+1}$ obtained with $\{(\boldsymbol{x}_j, y_j)\}_{j=1}^{m+1}$ in a batch manner is expressed as

$$\hat{f}_{m+1} = A_{m+1}^\dagger \boldsymbol{y}^{(m+1)}. \tag{5.67}$$

Let $\mathcal{L}(\{\varphi_j(\boldsymbol{x})\}_j)$ be the linear manifold spanned by $\{\varphi_j(\boldsymbol{x})\}_j$. Then we have the following incremental LMS learning method.

Theorem 5.11 (Incremental LMS learning) *The posterior LMS learning function $\hat{f}_{m+1}(\boldsymbol{x})$ is expressed by using an additional training example $(\boldsymbol{x}_{m+1}, y_{m+1})$ and the prior LMS learning function $\hat{f}_m(\boldsymbol{x})$ as follows.*

(a) When $K(\boldsymbol{x}, \boldsymbol{x}_{m+1}) \notin \mathcal{L}\left(\{K(\boldsymbol{x}, \boldsymbol{x}_j)\}_{j=1}^m\right)$,

$$\hat{f}_{m+1}(\boldsymbol{x}) = \hat{f}_m(\boldsymbol{x}) + \frac{y_{m+1} - \hat{f}_m(\boldsymbol{x}_{m+1})}{\|P_{\mathcal{N}(A_m)} K(\cdot, \boldsymbol{x}_{m+1})\|^2} P_{\mathcal{N}(A_m)} K(\boldsymbol{x}, \boldsymbol{x}_{m+1}), \tag{5.68}$$

where $P_{\mathcal{N}(A_m)}$ *denotes the orthogonal projection operator onto the null space of* A_m.

(b) When $K(\boldsymbol{x}, \boldsymbol{x}_{m+1}) \in \mathcal{L}\left(\{K(\boldsymbol{x}, \boldsymbol{x}_j)\}_{j=1}^m\right)$,

$$\hat{f}_{m+1}(\boldsymbol{x}) = \hat{f}_m(\boldsymbol{x}) + \frac{y_{m+1} - \hat{f}_m(\boldsymbol{x}_{m+1})}{1 + \langle (A_m^* A_m)^\dagger K(\cdot, \boldsymbol{x}_{m+1}), K(\cdot, \boldsymbol{x}_{m+1}) \rangle} (A_m^* A_m)^\dagger K(\boldsymbol{x}, \boldsymbol{x}_{m+1}). \tag{5.69}$$

A proof of Theorem 5.11 is provided in Section 5.8.6

Note that $\hat{f}_{m+1}(\boldsymbol{x})$ obtained by the above incremental LMS learning method exactly agrees with that obtained by batch LMS learning given by Eq.(5.67).

5.4.3 Two-stage sampling strategy

Based on the incremental LMS leaning method given in Theorem 5.11, we shall give a basic sampling strategy.

Let $J_b^{(m+1)}$ and $J_v^{(m+1)}$ be the variations in the bias and variance through the addition of a new training example $(\boldsymbol{x}_{m+1}, y_{m+1})$, respectively, i.e.,

$$J_b^{(m+1)} = \|\mathrm{E}_\epsilon \hat{f}_{m+1} - f\|^2 - \|\mathrm{E}_\epsilon \hat{f}_m - f\|^2, \tag{5.70}$$

$$J_v^{(m+1)} = \mathrm{E}_\epsilon \|\hat{f}_{m+1} - \mathrm{E}_\epsilon \hat{f}_{m+1}\|^2 - \mathrm{E}_\epsilon \|\hat{f}_m - \mathrm{E}_\epsilon \hat{f}_m\|^2. \tag{5.71}$$

Then the following lemma holds.

Lemma 5.12 $J_b^{(m+1)}$ *and* $J_v^{(m+1)}$ *are expressed as follows.*

(a) When $K(\boldsymbol{x}, \boldsymbol{x}_{m+1}) \notin \mathcal{L}\left(\{K(\boldsymbol{x}, \boldsymbol{x}_j)\}_{j=1}^m\right)$,

$$J_b^{(m+1)} = -\frac{\left|\langle f, P_{\mathcal{N}(A_m)} K(\cdot, \boldsymbol{x}_{m+1}) \rangle\right|^2}{\|P_{\mathcal{N}(A_m)} K(\cdot, \boldsymbol{x}_{m+1})\|^2}, \tag{5.72}$$

$$J_v^{(m+1)} = \sigma^2 \frac{1 + \langle (A_m^* A_m)^\dagger K(\cdot, \boldsymbol{x}_{m+1}), K(\cdot, \boldsymbol{x}_{m+1}) \rangle}{\|P_{\mathcal{N}(A_m)} K(\cdot, \boldsymbol{x}_{m+1})\|^2}. \tag{5.73}$$

(b) When $K(\boldsymbol{x}, \boldsymbol{x}_{m+1}) \in \mathcal{L}\left(\{K(\boldsymbol{x}, \boldsymbol{x}_j)\}_{j=1}^m\right)$,

$$J_b^{(m+1)} = 0, \tag{5.74}$$

$$J_v^{(m+1)} = -\sigma^2 \frac{\|(A_m^* A_m)^\dagger K(\cdot, \boldsymbol{x}_{m+1})\|^2}{1 + \langle (A_m^* A_m)^\dagger K(\cdot, \boldsymbol{x}_{m+1}), K(\cdot, \boldsymbol{x}_{m+1}) \rangle}. \tag{5.75}$$

A proof of Lemma 5.12 is given in Section 5.8.7.

Lemma 5.12 implies that the unknown learning target function $f(\boldsymbol{x})$ is included in Eq.(5.72) and the minimization of $J_v^{(m+1)}$ can be performed without knowing the learning target function $f(\boldsymbol{x})$ and the value of the noise variance σ^2. Note that Eq.(5.75) is equivalent to the active learning criterion used in the variance-only methods (see Section 5.5.5).

From Lemma 5.12, we have the following lemma.

Lemma 5.13 *For any additional training example $(\boldsymbol{x}_{m+1}, y_{m+1})$, the following relations hold.*

(a) When $K(\boldsymbol{x}, \boldsymbol{x}_{m+1}) \notin \mathcal{L}\left(\{K(\boldsymbol{x}, \boldsymbol{x}_j)\}_{j=1}^m\right)$,

$$J_b^{(m+1)} \leq 0 \quad \text{and} \quad J_v^{(m+1)} \geq 0. \tag{5.76}$$

(b) When $K(\boldsymbol{x}, \boldsymbol{x}_{m+1}) \in \mathcal{L}\left(\{K(\boldsymbol{x}, \boldsymbol{x}_j)\}_{j=1}^m\right)$,

$$J_b^{(m+1)} = 0 \quad \text{and} \quad J_v^{(m+1)} \leq 0. \tag{5.77}$$

Lemma 5.13 is clear from Lemma 5.12 since $(A_m^* A_m)^\dagger$ is positive semidefinite. Therefore, we omit the proof.

Lemma 5.13 states that an additional training example such that $K(\boldsymbol{x}, \boldsymbol{x}_{m+1}) \notin \mathcal{L}\left(\{K(\boldsymbol{x}, \boldsymbol{x}_j)\}_{j=1}^m\right)$ reduces or maintains the bias while it increases or maintains the variance. In contrast, an additional training example such that $K(\boldsymbol{x}, \boldsymbol{x}_{m+1}) \in \mathcal{L}\left(\{K(\boldsymbol{x}, \boldsymbol{x}_j)\}_{j=1}^m\right)$ maintains the bias while it reduces or maintains the variance.

As regards the bias, we have the following lemma.

Lemma 5.14 *The bias of $\hat{f}_m(\boldsymbol{x})$ vanishes if $\mathcal{L}\left(\{K(\boldsymbol{x}, \boldsymbol{x}_j)\}_{j=1}^m\right) = H$.*

A proof of Lemma 5.14 is given in Section 5.8.8.

Based on the above lemmas, let us consider the following *two-stage sampling scheme* (see Figure 5.8). We start from $m = 0$. In Stage 1, sample points such that $K(\boldsymbol{x}, \boldsymbol{x}_{m+1}) \notin \mathcal{L}\left(\{K(\boldsymbol{x}, \boldsymbol{x}_j)\}_{j=1}^m\right)$ are added for reducing the bias until it vanishes. In this case, the variance increases as shown in Lemma 5.13 (a), so the sample point that maximally suppresses the increase of the variance (i.e., minimizes $J_v^{(m+1)}$) is selected. Let μ be the dimension of H. Stage 1 ends if a sample point such that $K(\boldsymbol{x}, \boldsymbol{x}_{m+1}) \notin \mathcal{L}\left(\{K(\boldsymbol{x}, \boldsymbol{x}_j)\}_{j=1}^m\right)$ is added μ times, by which $\mathcal{L}\left(\{K(\boldsymbol{x}, \boldsymbol{x}_j)\}_{j=1}^\mu\right) \doteq H$ is attained (see Lemma 5.14). In Stage 2, sample

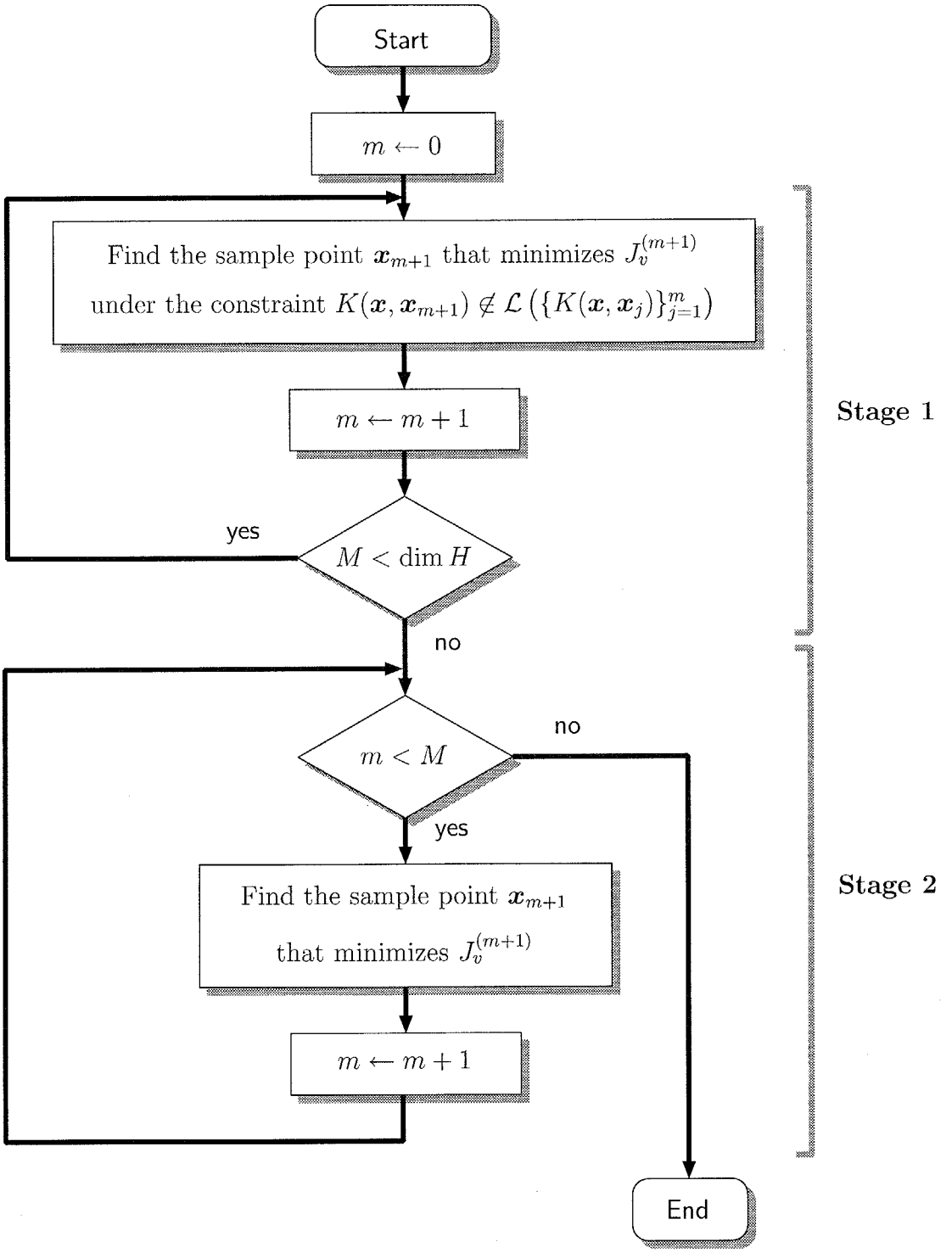


Figure 5.8: Two-stage sampling scheme.

points such that $K(\boldsymbol{x}, \boldsymbol{x}_{m+1}) \in \mathcal{L}\left(\{K(\boldsymbol{x}, \boldsymbol{x}_j)\}_{j=1}^m\right)$ are added for reducing the variance until the number of added sample points reaches M. The additional sample points such that $K(\boldsymbol{x}, \boldsymbol{x}_{m+1}) \in \mathcal{L}\left(\{K(\boldsymbol{x}, \boldsymbol{x}_j)\}_{j=1}^m\right)$ maintain the bias as shown in Lemma 5.13 (b). Hence, the bias remains zero throughout Stage 2. We select the sample point that maximizes the decrease of the variance (i.e., minimizes $J_v^{(m+1)}$). Since $\mathcal{L}\left(\{K(\boldsymbol{x}, \boldsymbol{x}_j)\}_{j=1}^m\right) = H$ is attained in Stage 1, all additional sample points satisfy $K(\boldsymbol{x}, \boldsymbol{x}_{m+1}) \in \mathcal{L}\left(\{K(\boldsymbol{x}, \boldsymbol{x}_j)\}_{j=1}^m\right)$ in Stage 2. This means that, in Stage 2, the constraint $K(\boldsymbol{x}, \boldsymbol{x}_{m+1}) \in \mathcal{L}\left(\{K(\boldsymbol{x}, \boldsymbol{x}_j)\}_{j=1}^m\right)$ does not have to be taken into account.

As a result, active learning problems in both stages become as follows.

Stage 1: Find the sample point $\hat{\boldsymbol{x}}_{m+1}$ that minimizes $J_v^{(m+1)}$ under the constraint of $K(\boldsymbol{x}, \boldsymbol{x}_{m+1}) \notin \mathcal{L}\left(\{K(\boldsymbol{x}, \boldsymbol{x}_j)\}_{j=1}^m\right)$.

Stage 2: Find the sample point $\hat{\boldsymbol{x}}_{m+1}$ that minimizes $J_v^{(m+1)}$.

In the rest of this section, we give two methods for finding the minimizer of $J_v^{(m+1)}$.

5.4.4 Minimization of $J_v^{(m+1)}$ by multi-point search

One of the naive methods for finding the minimizer of $J_v^{(m+1)}$ is *multi-point search*, i.e., a finite number of candidate points $\{\boldsymbol{x}_{m+1}^{(c)}\}_c$ are generated in the domain and the minimizer $\hat{\boldsymbol{x}}_{m+1}$ among the candidates is selected (Figure 5.9):

$$\hat{\boldsymbol{x}}_{m+1} = \underset{\boldsymbol{x}_{m+1} \in \{\boldsymbol{x}_{m+1}^{(c)}\}_c}{\operatorname{argmin}} J_v^{(m+1)}[\boldsymbol{x}_{m+1}] \tag{5.78}$$

Here, we show a practical calculation method of the proposed two-stage active learning algorithm by matrix operations.

Let $\{\varphi_p(\boldsymbol{x})\}_{p=1}^{\mu}$ be an orthonormal basis in H. Let B_m be an $m \times \mu$ matrix with the (j,p)-th element being $\varphi_p(\boldsymbol{x}_j)$:

$$[B_m]_{j,p} = \varphi_p(\boldsymbol{x}_j), \tag{5.79}$$

where $[\,\cdot\,]_{j,p}$ denotes the (j,p)-th element of a matrix. Let C_m be

$$C_m = B_m^* B_m. \tag{5.80}$$

C_m is the μ-dimensional matrix with the (p,p')-th element being $\sum_{m=1}^M \overline{\varphi_p(\boldsymbol{x}_m)}\varphi_{p'}(\boldsymbol{x}_m)$. Let G_m be a μ-dimensional matrix defined as

$$G_m = I_\mu - B_m^\dagger B_m. \tag{5.81}$$

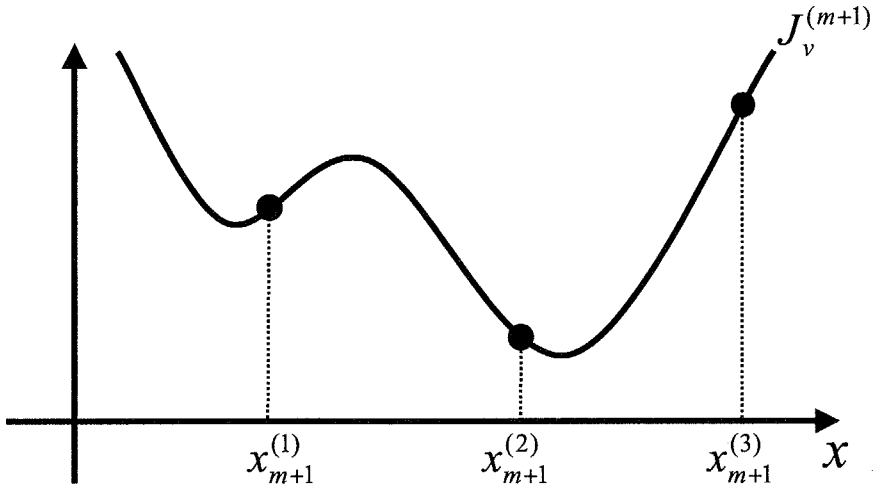


Figure 5.9: Multi-point search. Finite number of candidate points $\{\boldsymbol{x}_{m+1}^{(c)}\}_c$ are generated in the domain, and the minimizer among the candidates is selected. In the figure, $\boldsymbol{x}_{m+1}^{(2)}$ is selected.

Let d_{m+1} be a μ-dimensional vector with the p-th element being $\overline{\varphi_p(\boldsymbol{x}_{m+1})}$:

$$[d_{m+1}]_p = \overline{\varphi_p(\boldsymbol{x}_{m+1})}, \tag{5.82}$$

where $[\,\cdot\,]_p$ denotes the p-th element of a vector. Then the following corollary holds.

Corollary 5.15 (Calculation of $J_v^{(m+1)}$) *$J_v^{(m+1)}$ can be calculated as follows.*

(a) When $K(\boldsymbol{x}, \boldsymbol{x}_{m+1}) \notin \mathcal{L}\left(\{K(\boldsymbol{x}, \boldsymbol{x}_j)\}_{j=1}^m\right)$,

$$J_v^{(m+1)}[\boldsymbol{x}_{m+1}] = \sigma^2 \frac{1 + \langle C_m^\dagger d_{m+1}, d_{m+1}\rangle}{\langle G_m d_{m+1}, d_{m+1}\rangle}. \tag{5.83}$$

(b) When $K(\boldsymbol{x}, \boldsymbol{x}_{m+1}) \in \mathcal{L}\left(\{K(\boldsymbol{x}, \boldsymbol{x}_j)\}_{j=1}^m\right)$,

$$J_v^{(m+1)}[\boldsymbol{x}_{m+1}] = -\sigma^2 \frac{\|C_m^\dagger d_{m+1}\|^2}{1 + \langle C_m^\dagger d_{m+1}, d_{m+1}\rangle}. \tag{5.84}$$

A proof of Corollary 5.15 is given in Section 5.8.9.

Let us express the LMS learning function $\hat{f}_m(\boldsymbol{x})$ as

$$\hat{f}_m(\boldsymbol{x}) = \sum_{p=1}^{\mu} [\boldsymbol{w}^{(m)}]_p \varphi_p(\boldsymbol{x}), \tag{5.85}$$

where $\boldsymbol{w}^{(m)}$ is the μ-dimensional vector with the p-th element being a coefficient of $\varphi_p(\boldsymbol{x})$. Then the following corollary holds.

Corollary 5.16 (Incremental calculation of LMS learning function) *Let us express the posterior LMS learning function* $\hat{f}_{m+1}(\boldsymbol{x})$ *by using a* μ*-dimensional vector* $\boldsymbol{w}^{(m+1)}$ *as*

$$\hat{f}_{m+1}(\boldsymbol{x}) = \sum_{p=1}^{\mu} [\boldsymbol{w}^{(m+1)}]_p \varphi_p(\boldsymbol{x}). \tag{5.86}$$

Then $\boldsymbol{w}^{(m+1)}$ *can be calculated by using* $\boldsymbol{w}^{(m)}$ *as follows.*

(a) When $K(\boldsymbol{x}, \boldsymbol{x}_{m+1}) \notin \mathcal{L}\left(\{K(\boldsymbol{x}, \boldsymbol{x}_j)\}_{j=1}^{m}\right)$,

$$\boldsymbol{w}^{(m+1)} = \boldsymbol{w}^{(m)} + \frac{y_{m+1} - \langle \boldsymbol{w}^{(m)}, d_{m+1} \rangle}{\langle G_m d_{m+1}, d_{m+1} \rangle} G_m d_{m+1}. \tag{5.87}$$

(b) When $K(\boldsymbol{x}, \boldsymbol{x}_{m+1}) \in \mathcal{L}\left(\{K(\boldsymbol{x}, \boldsymbol{x}_j)\}_{j=1}^{m}\right)$,

$$\boldsymbol{w}^{(m+1)} = \boldsymbol{w}^{(m)} + \frac{y_{m+1} - \langle \boldsymbol{w}^{(m)}, d_{m+1} \rangle}{1 + \langle C_m^{\dagger} d_{m+1}, d_{m+1} \rangle} C_m^{\dagger} d_{m+1}. \tag{5.88}$$

A proof of Corollary 5.16 is provided in Section 5.8.10.

In the above corollaries, $J_v^{(m+1)}$ and $\boldsymbol{w}^{(m+1)}$ are calculated by using prior calculation results $C_m^{\dagger}$ and G_m (G_m is not used in Stage 2). Now we give their incremental calculation methods.

Lemma 5.17 (Incremental calculation of $C_m^{\dagger}$ and G_m) $C_{m+1}^{\dagger}$ *and* G_{m+1} *can be calculated by using* $C_m^{\dagger}$ *and* G_m *as follows.*

(a) When $K(\boldsymbol{x}, \boldsymbol{x}_{m+1}) \notin \mathcal{L}\left(\{K(\boldsymbol{x}, \boldsymbol{x}_j)\}_{j=1}^{m}\right)$,

$$\begin{aligned} C_{m+1}^{\dagger} = C_m^{\dagger} &+ \frac{1 + \langle C_m^{\dagger} d_{m+1}, d_{m+1} \rangle}{|\langle G_m d_{m+1}, d_{m+1} \rangle|^2} G_m d_{m+1} \otimes \overline{G_m d_{m+1}} \\ &- \frac{C_m^{\dagger} d_{m+1} \otimes \overline{G_m d_{m+1}} + G_m d_{m+1} \otimes \overline{C_m^{\dagger} d_{m+1}}}{\langle G_m d_{m+1}, d_{m+1} \rangle}, \end{aligned} \tag{5.89}$$

$$G_{m+1} = G_m - \frac{G_m d_{m+1} \otimes \overline{G_m d_{m+1}}}{\langle G_m d_{m+1}, d_{m+1} \rangle}. \tag{5.90}$$

(b) When $K(\boldsymbol{x}, \boldsymbol{x}_{m+1}) \in \mathcal{L}\left(\{K(\boldsymbol{x}, \boldsymbol{x}_j)\}_{j=1}^{m}\right)$,

$$C_{m+1}^{\dagger} = C_m^{\dagger} - \frac{C_m^{\dagger} d_{m+1} \otimes \overline{C_m^{\dagger} d_{m+1}}}{1 + \langle C_m^{\dagger} d_{m+1}, d_{m+1} \rangle}. \tag{5.91}$$

A proof of Lemma 5.17 is given in Section 5.8.11.

Based on the above discussion, a complete algorithm of the two-stage active learning by multi-point search is described in Figure 5.10.

$\mu \longleftarrow \dim H$, $\boldsymbol{w}^{(0)} \longleftarrow 0$, $C_0^\dagger \longleftarrow 0$, $G_0 \longleftarrow I_\mu$;
for $m = 0, 1, \ldots, \mu - 1$ {
  Generate candidate points $\{\boldsymbol{x}_{m+1}^{(c)}\}_c$ in the domain
    such that $K(\boldsymbol{x}, \boldsymbol{x}_{m+1}^{(c)}) \notin \mathcal{L}\left(\{K(\boldsymbol{x}, \boldsymbol{x}_j)\}_{j=1}^m\right)$;
  $\hat{\boldsymbol{x}}_{m+1} \leftarrow \mathrm{argmin}_{\boldsymbol{x}_{m+1} \in \{\boldsymbol{x}_{m+1}^{(c)}\}_c} J_v^{(m+1)}[\boldsymbol{x}_{m+1}]$
    where $J_v^{(m+1)}$ is given by Eq.(5.83);
  Sample y_{m+1} at $\hat{\boldsymbol{x}}_{m+1}$;
  Carry out incremental LMS learning with $(\hat{\boldsymbol{x}}_{m+1}, y_{m+1})$ by Eq.(5.87);
  Calculate $C_{m+1}^\dagger$ and G_{m+1} by Eqs.(5.89) and (5.90);
}
for $m = \mu, \mu + 1, \ldots, M - 1$ {
  Generate candidate points $\{\boldsymbol{x}_{m+1}^{(c)}\}_c$ in the domain;
  $\hat{\boldsymbol{x}}_{m+1} \leftarrow \mathrm{argmin}_{\boldsymbol{x}_{m+1} \in \{\boldsymbol{x}_{m+1}^{(c)}\}_c} J_v^{(m+1)}[\boldsymbol{x}_{m+1}]$
    where $J_v^{(m+1)}$ is given by Eq.(5.84);
  Sample y_{m+1} at $\hat{\boldsymbol{x}}_{m+1}$;
  Carry out incremental LMS learning with $(\hat{\boldsymbol{x}}_{m+1}, y_{m+1})$ by Eq.(5.88);
  Calculate $C_{m+1}^\dagger$ by Eq.(5.91);
}

Figure 5.10: Algorithm of two-stage active learning with multi-point search.

5.4.5 Minimization of $J_v^{(m+1)}$ by gradient-descent search

The multi-point search method is easy to implement. However, if the dimension L of the input vector $\boldsymbol{x}$ is very large, many candidate points may be required for finding a better sampling location by multi-point search. One of the measures is to use the *gradient-descent* method for finding a local minimum of $J_v^{(m+1)}$. That is, starting from some initial value, $\boldsymbol{x}_{m+1}$ is updated until some convergence criterion holds as

$$\boldsymbol{x}_{m+1} \longleftarrow \boldsymbol{x}_{m+1} - \gamma'' \nabla J_v^{(m+1)}[\boldsymbol{x}_{m+1}], \tag{5.92}$$

where γ'' is a small positive constant and ∇ is an operator defined as

$$\nabla = \left(\frac{\partial}{\partial \xi^{(1)}}, \frac{\partial}{\partial \xi^{(2)}}, \ldots, \frac{\partial}{\partial \xi^{(L)}}\right)^\top, \tag{5.93}$$

where

$$\boldsymbol{x} = (\xi^{(1)}, \xi^{(2)}, \ldots, \xi^{(L)})^\top. \tag{5.94}$$

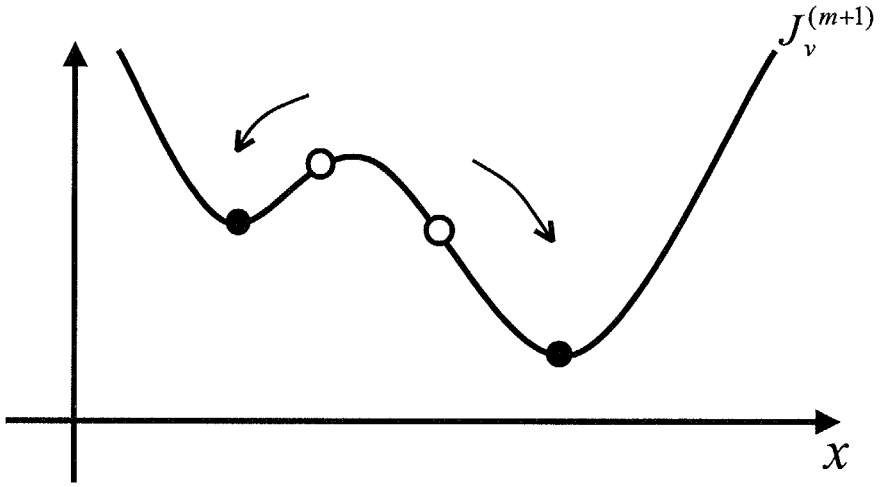


Figure 5.11: Gradient-descent search. Initial points are denoted by '○'. They descend along the slope and converge to local minima (denoted by '●').

This idea of gradient-descent search is illustrated in Figure 5.11.

Let $h_{m+1}^{(l)}$ be a μ-dimensional vector with the p-th element being $\frac{\partial}{\partial \xi^{(l)}} \overline{\varphi_p(\boldsymbol{x}_{m+1})}$:

$$[h_{m+1}]_p = \frac{\partial}{\partial \xi^{(l)}} \overline{\varphi_p(\boldsymbol{x}_{m+1})}. \tag{5.95}$$

Then the following corollary holds.

Corollary 5.18 (Gradient of $J_v^{(m+1)}$) *The gradient of $J_v^{(m+1)}$ is expressed as follows.*

(a) When $K(\boldsymbol{x}, \boldsymbol{x}_{m+1}) \notin \mathcal{L}\left(\{K(\boldsymbol{x}, \boldsymbol{x}_j)\}_{j=1}^m\right)$,

$$\begin{aligned}\frac{\partial}{\partial \xi^{(l)}} J_v^{(m+1)}[\boldsymbol{x}_{m+1}] = 2\sigma^2 \Big(& \langle C_m^\dagger d_{m+1}, h_{m+1} \rangle \langle G_m d_{m+1}, d_{m+1} \rangle \\ & - \Big(1 + \langle C_m^\dagger d_{m+1}, d_{m+1} \rangle\Big) \langle G_m d_{m+1}, h_{m+1} \rangle \Big) \\ & \Big/ \Big(\langle (I_\mu - C_m^\dagger C_m) d_{m+1}, d_{m+1} \rangle \Big)^2 .\end{aligned} \tag{5.96}$$

(b) When $K(\boldsymbol{x}, \boldsymbol{x}_{m+1}) \in \mathcal{L}\left(\{K(\boldsymbol{x}, \boldsymbol{x}_j)\}_{j=1}^m\right)$,

$$\begin{aligned}\frac{\partial}{\partial \xi^{(l)}} J_v^{(m+1)}[\boldsymbol{x}_{m+1}] = -2\sigma^2 \Big(& \langle C_m^\dagger C_m^\dagger d_{m+1}, h_{m+1} \rangle (1 + \langle C_m^\dagger d_{m+1}, d_{m+1} \rangle) \\ & - \|C_m^\dagger d_{m+1}\|^2 \langle C_m^\dagger d_{m+1}, h_{m+1} \rangle \Big) \\ & \Big/ \Big(1 + \langle C_m^\dagger d_{m+1}, d_{m+1} \rangle\Big)^2 .\end{aligned} \tag{5.97}$$

Corollary 5.18 is clear from Corollary 5.15 and the derivative of a fraction:

$$\frac{\partial}{\partial \xi}\left(\frac{f}{g}\right) = \frac{\frac{\partial f}{\partial \xi}g - \frac{\partial g}{\partial \xi}f}{g^2}. \tag{5.98}$$

Therefore, we omit the proof.

5.5 Comparison with existing active learning techniques

In this section, the proposed active learning methods are compared with existing active learning methods.

5.5.1 Overview of existing active learning techniques and placement of proposed methods

Various active learning methods have been proposed so far. Here, we categorize them into seven groups depending on the type of the error measure (Figure 5.12): 1. D-optimal design, 2. minimax design, 3. A-optimal design, 4. variance-only design, 5. bias-only design, 6. two-stage design, and 7. Bayesian statistics based design.

Since the batch active learning method proposed in Section 5.3 minimize the variance under the constraint of the bias being zero, it can be regarded as a variance-only design. The incremental active learning method proposed in Section 5.4 reduces the bias and variance in two stages. Therefore, it is a two-stage design.

Based on the above categorization, we shall review the existing active learning methods, and investigate the relation to the proposed methods.

5.5.2 D-optimal design

In mathematical statistics, the *D-optimal design* has been thoroughly studied (Kiefer [60], Kiefer & Wolfowitz [61], Box & Hunter [18], Fedorov [34]). The D-optimal design $\hat{\mathcal{X}}$ minimizes the determinant of the dispersion matrix:

$$\hat{\mathcal{X}} = \operatorname*{argmin}_{\mathcal{X}} \det(C_{\mathcal{X}}^{-1}), \tag{5.99}$$

where $\det(\cdot)$ denotes the determinant of a matrix and $C_{\mathcal{X}}$ is given by Eq.(5.34). One of the advantages of the D-optimal design is that it is invariant under all affine transformations in the input space (Kiefer [60]). In spite of the preferable property, the D-optimal design does

1. **D-optimal design**
   - Kiefer (1959)
   - Box and Hunter (1965)
   - Fedorov (1972)
2. **Minimax design**
   - Kiefer and Wolfowitz (1960)
   - Fedorov (1972)
3. **A-optimal design**
   - Kiefer (1959)
   - Fedorov (1972)
4. **Variance-only design (Q-optimal design)**
   - Cohn (1996)
   - Cohn, Ghahramani, and Jordan (1996)
   - Fukumizu (2000)
   - Batch active learning method proposed in Section 5.3
5. **Bias-only design**
   - Cohn (1997)
   - Vijayakumar and Ogawa (1999)
   - Yue and Hickernell (1999)
6. **Two-stage design**
   - Vijayakumar, Sugiyama, and Ogawa (1998)
   - Incremental active learning method proposed in Section 5.4
7. **Bayesian statistics based design**
   - MacKay (1992)

Figure 5.12: Categorization of active learning methods.

not directly evaluate the generalization error itself. Therefore, the optimal generalization capability is not guaranteed.

5.5.3 Minimax design

The *minimax design* $\hat{\mathcal{X}}$ minimizes the maximum variance (Kiefer & Wolfowitz [61]):

$$\hat{\mathcal{X}} = \operatorname*{argmin}_{\mathcal{X}} \max_{\boldsymbol{u}} \mathrm{E}_{\epsilon} \left| \hat{f}_{\mathcal{X}}(\boldsymbol{u}) - \mathrm{E}_{\epsilon} \hat{f}_{\mathcal{X}}(\boldsymbol{u}) \right|^2 . \tag{5.100}$$

Kiefer and Wolfowitz [61] showed that when the noise variance is the same magnitude all over the domain, the minimax design agrees with the D-optimal design. However, the minimax design does not guarantee the optimal generalization capability since it does not directly evaluate the generalization error itself.

5.5.4 A-optimal design

The *A-optimal design* $\hat{\mathcal{X}}$ minimizes the trace of the dispersion matrix (Kiefer [60], Fedorov [34]).

$$\hat{\mathcal{X}} = \operatorname*{argmin}_{\mathcal{X}} \mathrm{tr} C_{\mathcal{X}}^{-1}, \tag{5.101}$$

where $C_{\mathcal{X}}$ is given by Eq.(5.34). It is known that the A-optimal design is also D-optimal if $C_{\mathcal{X}}^{-2} = k C_{\mathcal{X}}^{-1}$ where k is some constant (Fedorov [34]). However, the A-optimal design does not guarantee the optimal generalization capability since it does not directly evaluate the generalization error itself.

5.5.5 Variance-only design

The *variance-only design* $\hat{\mathcal{X}}$ minimizes the variance (Cohn [25], Cohn *et al.* [27], Fukumizu [38]):

$$\hat{\mathcal{X}} = \operatorname*{argmin}_{\mathcal{X}} \mathrm{E}_{\epsilon} \| \hat{f}_{\mathcal{X}} - \mathrm{E}_{\epsilon} \hat{f}_{\mathcal{X}} \|^2. \tag{5.102}$$

The variance-only design is also referred to as the *Q-optimal design*, and it agrees with the A-optimal design if the basis functions $\{\varphi_p(\boldsymbol{x})\}_{p=1}^{\mu}$ are orthonormal (Fedorov [34]).

As shown in Eq.(5.20), the generalization error consists of the bias and variance. Therefore, variance-only methods are effective only when the bias is zero or small enough to be neglected. Since the existing variance-only methods do not take the bias into account, the optimal generalization capability is not theoretically guaranteed.

The batch active learning method proposed in Section 5.3 can be regarded as a variance-only method. However, different from the existing methods, the proposed batch active learning method *does* guarantee the optimal generalization capability since the bias always vanishes.

5.5.6 Bias-only design

Vijayakumar and Ogawa [143] gave a necessary and sufficient condition of sample points for providing the optimal generalization capability in the absence of noise. In their paper, *Wiener learning* is adopted as a learning criterion (see e.g. Ogawa & Oja [94]), and the generalization error (which corresponds to the bias in the presence of noise) is measured by

$$\mathrm{E}_f \|\hat{f}_{\mathcal{X}} - f\|^2, \tag{5.103}$$

where E_f denotes the expectation over the learning target function $f(\boldsymbol{x})$. In Wiener learning, the correlation operator R of the learning target function $f(\boldsymbol{x})$ should be available:

$$R = \mathrm{E}_f \left(f \otimes \overline{f} \right). \tag{5.104}$$

Although the optimality condition was derived, design methods of sample points that satisfy the condition are not given yet.

Cohn [26] used resampling methods such as the *bootstrapping* (Efron & Tibshirani [33]) and the *cross-validation* (e.g. Mosteller & Wallace [78], Allen [6]) for estimating the bias, and proposed an active learning method for reducing the bias.

Yue and Hickernell [149] showed an upper bound of the generalization error and derived an active learning criterion for minimizing the upper bound. This criterion includes an unknown controlling parameter of the trade-off between the bias and variance, so the optimal solution can not be obtained.

Even though the bias-only methods are experimentally shown to outperform the variance-only methods (Cohn [26], Yue & Hickernell [149]), their effectiveness is not theoretically guaranteed since the variance is not taken into account.

5.5.7 Two-stage design

Vijayakumar *et al.* [144] proposed combining the bias-only and variance-only methods in two stages. This idea forms the basis of the two-stage active learning method proposed in Section 5.4.

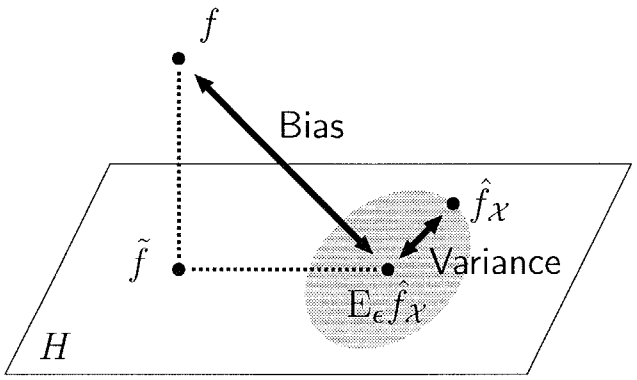


Figure 5.13: Interpretation of assumptions in variance-only and proposed two-stage active learning methods. Let $\tilde{f}$ be the best approximation to f in H. Variance-only active learning methods assume that $f = \tilde{f}$ and $\tilde{f} = \mathrm{E}_{\epsilon}\hat{f}_{\mathcal{X}}$. Namely, f belongs to H and the expectation of $\hat{f}_{\mathcal{X}}$ over the noise agrees with $\tilde{f}$. In contrast, the proposed method only assumes that $f \in H$. The difference between $\tilde{f}(= f)$ and $\mathrm{E}_{\epsilon}\hat{f}_{\mathcal{X}}$ is explicitly evaluated in Stage 1.

The two-stage active learning method proposed in Section 5.4 can be regarded as an extension of the variance-only methods proposed by Cohn [25], Cohn *et al.* [27], and Fukumizu [38]. In the variance-only methods, the bias is assumed to be zero. The assumption is equivalent to that f belongs to H and the expectation of $\hat{f}_{\mathcal{X}}$ over the noise agrees with f. In contrast, the condition assumed in the two-stage active learning method proposed in Section 5.4 is only $f \in H$. The difference between f and $\mathrm{E}_{\epsilon}\hat{f}_{\mathcal{X}}$ is explicitly evaluated in Stage 1. The interpretation is summarized in Figure 5.13.

5.5.8 Bayesian statistics based design

Within the framework of Bayesian statistics, MacKay [70] proposed an active learning method. He used the Gaussian approximation in the derivation, and the resulting criterion is essentially the same as the variance-only method.

5.6 Computer simulations

In this section, the effectiveness of the proposed active learning methods is demonstrated through computer simulations.

5.6.1 One-dimensional trigonometric polynomial model

First, we consider the case where the dimension L of the input vector $\boldsymbol{x}$ is 1.

Let H be spanned by the functions

$$\left\{1, \sqrt{2}\sin nx, \sqrt{2}\cos nx\right\}_{n=1}^{100} \tag{5.105}$$

defined on $[-\pi, \pi]$, and the inner product is defined as

$$\langle f, g\rangle = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(x)\overline{g(x)}dx, \tag{5.106}$$

i.e., H is a trigonometric polynomial space of order 100 (see Section 3.3.1). Note that the dimension of H is 201. Let the learning target function $f(x)$ be

$$\begin{aligned} f(x) = {} & 2\sqrt{2}\sin x + 2\sqrt{2}\cos x + \frac{\sqrt{2}}{2}\sin 2x + \sqrt{2}\cos 2x \\ & - 2\sqrt{2}\sin 3x + 2\sqrt{2}\cos 3x - \frac{\sqrt{2}}{10}\sin 4x + \frac{\sqrt{2}}{2}\cos 5x. \end{aligned} \tag{5.107}$$

Let the noise ϵ_m be independently subject to the same normal distribution with mean zero and variance $\sigma^2 = 1$:

$$\epsilon_m \sim N(0, 1). \tag{5.108}$$

In this case, the noise covariance matrix Q is given as

$$Q = I_M. \tag{5.109}$$

Let us consider the following sampling schemes.

(a) Optimal sampling: Sample points are determined following Corollary 5.6.

(b) Two-stage active learning: Sample points are determined by the two-stage active learning method with multi-point search. 3 randomly created candidate points are used for multi-point search.

(c) Variance-only method: Eq.(5.75) is adopted as the active learning criterion. Sample points are determined by multi-point search with 3 randomly created candidate points.

(d) Passive learning: Sample points are randomly created.

Note that the sampling scheme (a) is a global optimal method while the sampling schemes (b) and (c) are greedy optimal methods.

Figure 5.14 displays the values of the bias, variance, and generalization error J_G through the addition of training examples. The horizontal axis denotes the number of training examples while the vertical axes denote the values of the bias, variance, and generalization error. The dotted curve in the bottom graph shows the generalization error by the sampling scheme (a). The solid, dashed, and dash-dotted curves denote the mean values of 100 trials by the sampling schemes (b), (c), and (d), respectively.

For the sampling scheme (b) shown by the dashed curve, $m \leq 201$ $(= \dim H)$ corresponds to Stage 1 and $m > 201$ corresponds to Stage 2, where m is the number of training examples. The bias decreases and the variance increases in Stage 1, and the bias remains zero and the variance decreases in Stage 2. This phenomenon is in good agreement with Lemma 5.13.

In Stage 1, the sampling scheme (b) suppresses the increase of the variance more efficiently than the sampling schemes (c) and (d). In Stage 2, the sampling schemes (b) and (c) suppress the variance more efficiently than the sampling scheme (d). As a result, the sampling scheme (b) gives better generalization capability than the sampling schemes (c) and (d) with a small number of training examples. Note that the sampling scheme (a) gives the optimal generalization capability.

The sampling schemes (b) and (c) can be applied to any Hilbert spaces while the sampling scheme (a) is restricted to the trigonometric polynomial space. This simulation suggests that when the model is the trigonometric polynomial space, the sampling scheme (a) can be applied and it gives the optimal generalization capability. Otherwise, the sampling scheme (b) seems to work well.

5.6.2 Multi-dimensional polynomial model

Now we consider the case where the sample points are multi-dimensional.

Let the dimension L of the input vector $\boldsymbol{x}$ be 4:

$$\boldsymbol{x} = (\xi^{(1)}, \xi^{(2)}, \xi^{(3)}, \xi^{(4)})^\top. \tag{5.110}$$

Let H be spanned by the functions

$$\left\{ \prod_{l=1}^{4} \left(\xi^{(l)}\right)^{n_l} \;\middle|\; n_l = 0, 1, 2 \quad \text{for } l = 1, 2, 3, 4 \right\}, \tag{5.111}$$

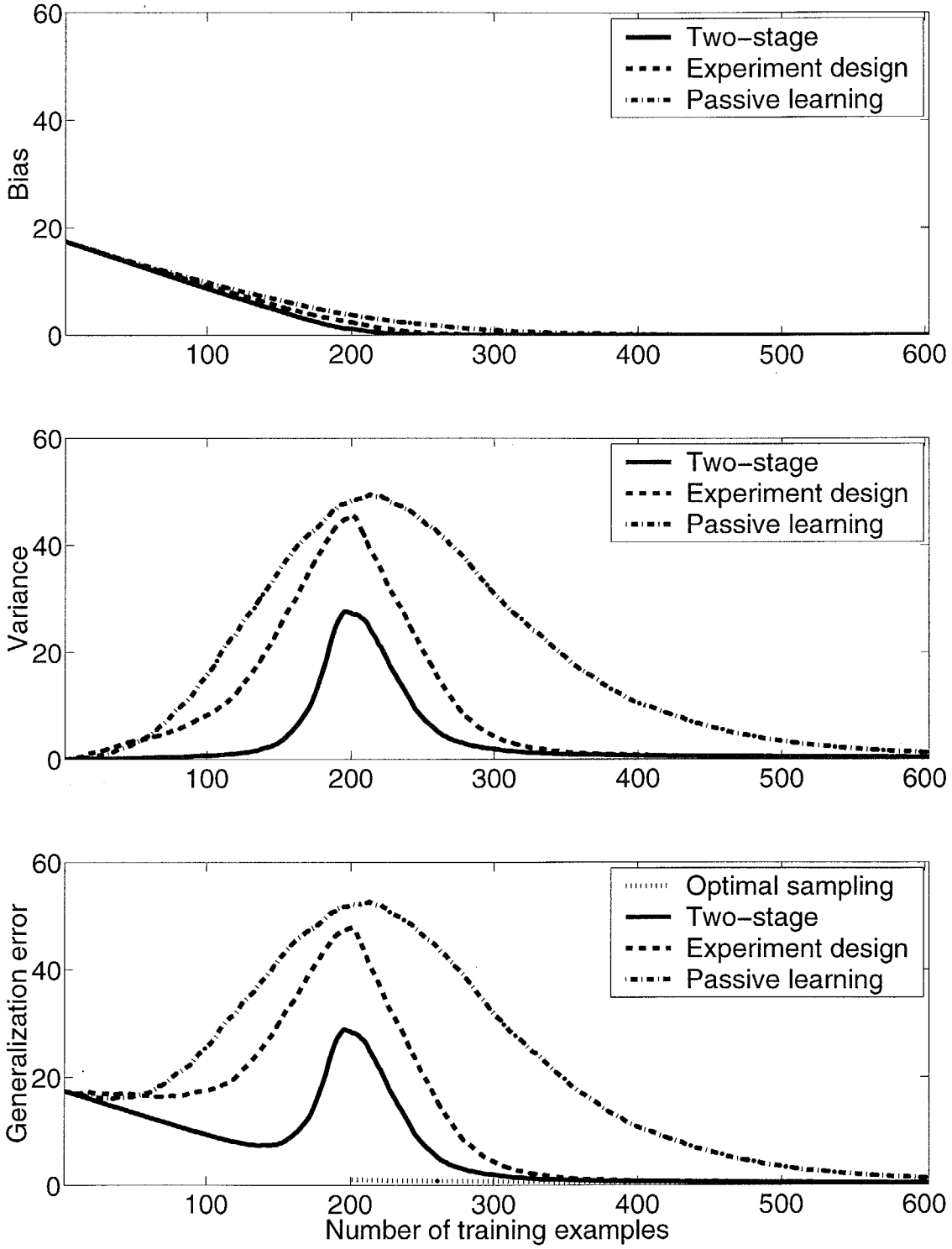


Figure 5.14: Results of active learning simulation for one-dimensional trigonometric polynomial space.

and the inner product is defined as

$$\langle f, g \rangle = \int_{-1}^{1} \int_{-1}^{1} \int_{-1}^{1} \int_{-1}^{1} f(\boldsymbol{x}) \overline{g(\boldsymbol{x})} d\xi^{(1)} d\xi^{(2)} \xi^{(3)} d\xi^{(4)}, \tag{5.112}$$

i.e., H is a polynomial space of order $(2, 2, 2, 2)$ (see Section 3.3.2). Note that the dimension of H is $3^4 = 81$. Let the learning target function $f(\boldsymbol{x})$ be

$$f(\boldsymbol{x}) = \frac{225}{16}\xi_1^2\xi_4^2 - \frac{63\sqrt{5}}{8}\xi_2^2\xi_3\xi_4 - \frac{75}{16}\xi_1^2 + \frac{9}{2}\xi_1\xi_3 + \frac{21\sqrt{5}}{8}\xi_3\xi_4 - \frac{75}{16}\xi_4^2 + \frac{61}{16}. \tag{5.113}$$

Let the noise ϵ_m be independently subject to the same normal distribution with mean zero and variance $\sigma^2 = 1$:

$$\epsilon_m \sim N(0, 1). \tag{5.114}$$

In this case, the noise covariance matrix Q is given as

$$Q = I_M. \tag{5.115}$$

Let us consider the following sampling schemes.

(a) **Two-stage active learning with multi-point search:** Sample points are determined by the two-stage active learning method with multi-point search. 3 randomly created candidate points are used for multi-point search.

(b) **Two-stage active learning with gradient-descent search:** Sample points are determined by the two-stage active learning method with gradient-descent search. 3 randomly created initial points are used for gradient-descent search.

(c) **Passive learning:** Sample points are randomly created.

The values of the bias, variance, and generalization error J_G through the addition of training examples are displayed in Figure 5.15. The horizontal axis denotes the number of training examples while the vertical axes denote the values of the bias, variance, and generalization error. The solid, dashed, and dash-dotted curves show the mean values of 100 trials by sampling schemes (a), (b) and (c), respectively. $m \leq 81$ corresponds to Stage 1 while $m > 81$ corresponds to Stage 2, where m is the number of training examples.

This simulation again shows that higher levels of the generalization capability can be acquired by the proposed two-stage method with a small number of training examples. Especially, the performance of the gradient-descent search method is excellent, although it is computationally expensive.

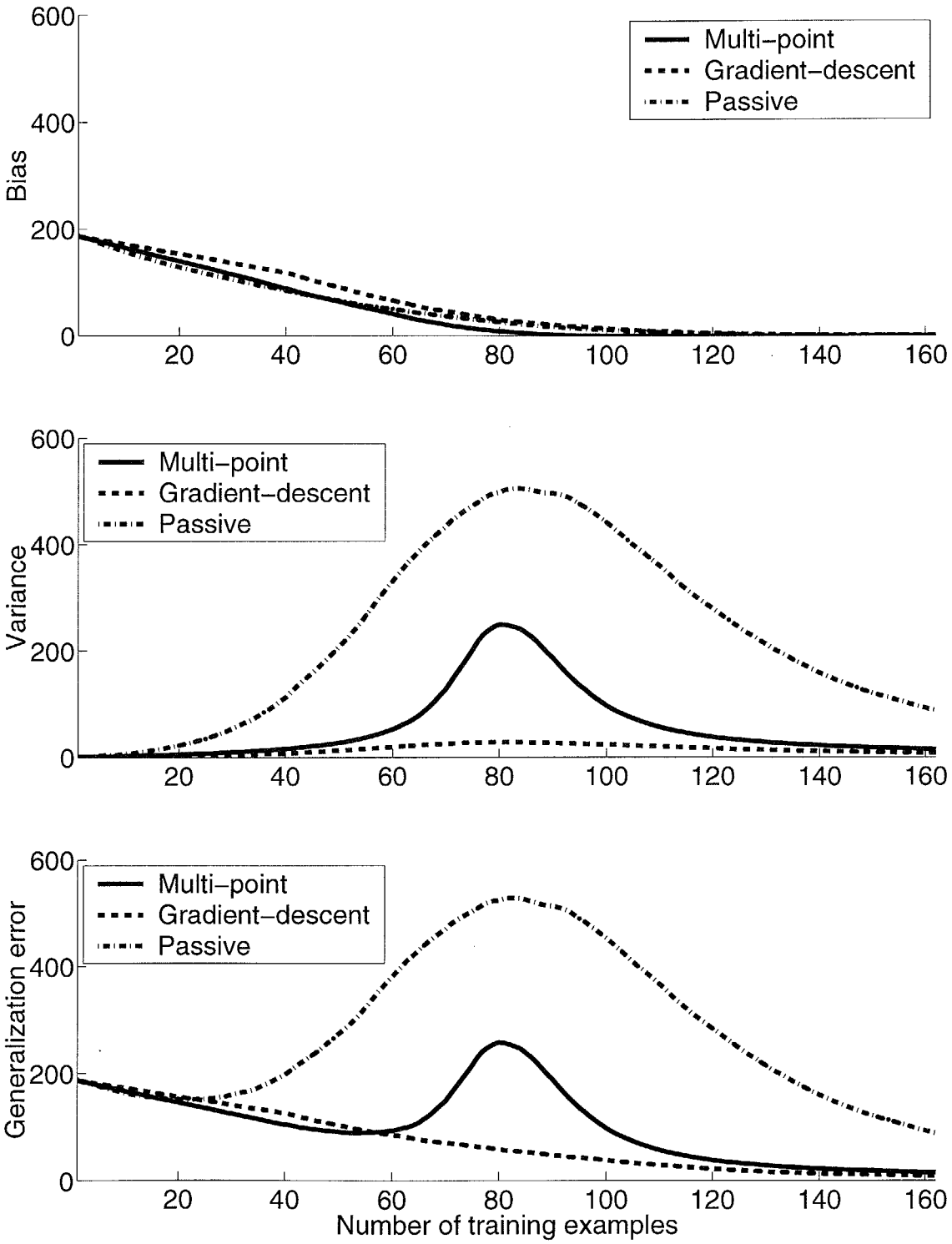


Figure 5.15: Results of active learning simulation for four-dimensional polynomial space.

5.7 Pseudo orthonormal bases

In Section 5.3.2, the necessary and sufficient condition for the optimal generalization capability (see Theorem 5.2) was characterized by using the properties of pseudo orthonormal bases (PONBs). PONBs are a special type of pseudo orthogonal bases (POBs). In this section, we briefly review the concept of POBs and PONBs, and show their fundamental properties.

Let H be a finite μ-dimensional Hilbert space and M be an integer larger than or equal to μ:

$$M \geq \mu. \tag{5.116}$$

Then POBs are defined as follows.

Definition 5.19 *(Ogawa & Iijima [92]) A set $\{\phi_m\}_{m=1}^M$ of elements in H is called a POB if any f in H is expressed as*

$$f = \sum_{m=1}^{M} \langle f, \phi_m \rangle \phi_m. \tag{5.117}$$

The concept of POBs is an extension of orthonormal bases (ONBs) to linearly dependent over-complete systems. It is clear that a POB is reduced to an ONB in H if M is equal to the dimension of H. POBs and their extension, pseudo biorthogonal bases (Ogawa [86][91]), have been successfully applied to various real world problems including signal restoration (Ogawa [87][91]), computerized tomography (Ogawa & Kumazawa [93]), neural network learning (Ogawa [90]), and robust construction of neural networks (Nakazawa & Ogawa [84], Iwaki *et al.* [56]). An example of POBs is given as follows.

Example 5.20 (Example of POBs) *(Ogawa & Iijima [92]) Let H be the two-dimensional unitary space $\mathbf{C}^2$ and $M = 3$. Then the following set $\{\phi_m\}_{m=1}^3$ forms a POB in H for $0 \leq \theta \leq 1$:*

$$\phi_1 = \frac{1}{\sqrt{2}}(1, -\theta)^\top, \tag{5.118}$$

$$\phi_2 = \frac{1}{\sqrt{2}}(1, \theta)^\top, \tag{5.119}$$

$$\phi_3 = \frac{1}{\sqrt{2}}(0, \sqrt{2(1-\theta^2)})^\top. \tag{5.120}$$

$\{\phi_m\}_{m=1}^3$ with $\theta = \frac{1}{\sqrt{2}}$ is illustrated in Figure 5.16. The following proposition shows basic characteristics of POBs.

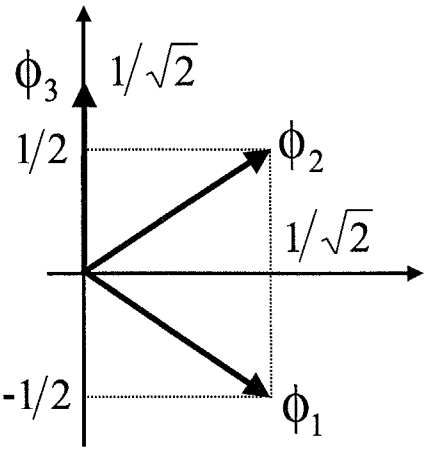


Figure 5.16: Example of POBs given in Example 5.20 with $\theta = \frac{1}{\sqrt{2}}$, i.e., $\phi_1 = (\frac{1}{\sqrt{2}}, -\frac{1}{2})^\top$, $\phi_2 = (\frac{1}{\sqrt{2}}, \frac{1}{2})^\top$, and $\phi_3 = (0, \frac{1}{\sqrt{2}})$.

Proposition 5.21 *(Ogawa & Iijima [92]) The following conditions are mutually equivalent.*

1. *A set* $\{\phi_m\}_{m=1}^M$ *is a POB in* H.
2. $\|f\|^2 = \sum_{m=1}^{M} |\langle f, \phi_m \rangle|^2$ for any $f \in H$.
3. $\langle f, g \rangle = \sum_{m=1}^{M} \langle f, \phi_m \rangle \overline{\langle g, \phi_m \rangle}$ for any $f, g \in H$.

Condition 2 implies that a POB is a *tight frame with frame bound one* (Daubechies [29]) or a *normalized tight frame* (Frank & Larson [35]) in the *frame* terminology. When M is equal to the dimension of H, Conditions 2 and 3 are reduced to *Parseval's equalities.*

Now let us consider a finite M-dimensional Hilbert space H'. Let a set $\{\varphi'_m\}_{m=1}^M$ be an ONB in H' and Y be an operator defined as

$$Y = \sum_{m=1}^{M} (\varphi'_m \otimes \overline{\phi_m}) . \tag{5.121}$$

Then the following proposition holds.

Proposition 5.22 *(Ogawa & Iijima [92]) The following conditions are mutually equivalent.*

1. *A set* $\{\phi_m\}_{m=1}^M$ *is a POB in* H.

2. $Y^*Y = I_H$, *where* I_H *is the identity operator on* H.

3. $\|Yf\| = \|f\|$ for any $f \in H$.

4. $\langle Yf, Yg \rangle = \langle f, g \rangle$ for any $f, g \in H$.

It follows from Condition 2 that

$$\sum_{m=1}^{M} \|\phi_m\|^2 = \mathrm{tr}\left(\sum_{m=1}^{M} (\phi_m \otimes \overline{\phi_m}) \right) = \mathrm{tr} Y^*Y = \mathrm{tr} I_H = \mu, \tag{5.122}$$

where μ is the dimension of H. Condition 3 means that the operator Y is an *isometry* (see Section 2.2.3). From these properties, we have the following construction method of POBs.

Proposition 5.23 *(Ogawa & Iijima [92]) Let* Y *be an isometry from* H *to* H' *and a set* $\{\varphi'_m\}_{m=1}^M$ *be an ONB in* H'. *If we let*

$$\phi_m = Y^* \varphi'_m \text{ for } m = 1, 2, \ldots, M, \tag{5.123}$$

then a set $\{\phi_m\}_{m=1}^M$ *becomes a POB in* H.

Note that all POBs can be constructed by changing Y with a fixed ONB $\{\varphi'_m\}_{m=1}^M$ or by changing $\{\varphi'_m\}_{m=1}^M$ with a fixed Y.

If a set $\{\phi_m\}_{m=1}^M$ is a POB and

$$\|\phi_1\| = \|\phi_2\| = \cdots = \|\phi_M\|, \tag{5.124}$$

then the set $\{\phi_m\}_{m=1}^M$ is called a *pseudo orthonormal basis* (PONB). In this case, it follows from Eq.(5.122) that

$$\|\phi_m\| = \sqrt{\frac{\mu}{M}} \text{ for } m = 1, 2, \ldots, M. \tag{5.125}$$

An example of PONBs is given as follows.

Example 5.24 (Example of PONBs) *(Ogawa & Iijima [92]) Let* H *be the two-dimensional unitary space* $\mathbf{C}^2$ *and* $M = 4$. *Then the following set* $\{\phi_m\}_{m=1}^4$ *forms a*

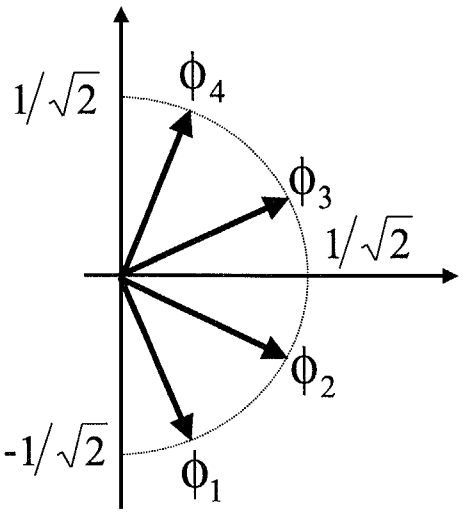


Figure 5.17: Example of PONBs given in Example 5.24 with $\theta = \frac{1}{2}$, i.e., $\phi_1 = \sqrt{\frac{2}{5}}(\frac{1}{2}, -1)^\top$, $\phi_2 = \sqrt{\frac{2}{5}}(1, -\frac{1}{2})^\top$, $\phi_3 = \sqrt{\frac{2}{5}}(1, \frac{1}{2})^\top$, and $\phi_4 = \sqrt{\frac{2}{5}}(\frac{1}{2}, 1)^\top$.

POB in H for $0 \leq \theta \leq 1$:

$$\phi_1 = \frac{1}{\sqrt{2(1+\theta^2)}}(\theta, -1)^\top, \tag{5.126}$$

$$\phi_2 = \frac{1}{\sqrt{2(1+\theta^2)}}(1, -\theta)^\top, \tag{5.127}$$

$$\phi_3 = \frac{1}{\sqrt{2(1+\theta^2)}}(1, \theta)^\top, \tag{5.128}$$

$$\phi_4 = \frac{1}{\sqrt{2(1+\theta^2)}}(\theta, 1)^\top. \tag{5.129}$$

$\{\phi_m\}_{m=1}^4$ with $\theta = \frac{1}{2}$ is illustrated in Figure 5.17.

Finally, we show a construction method of PONBs that plays an important role in the proof of Theorem 5.9.

Theorem 5.25 *Let $M = k\mu$ where k is a positive integer and μ is the dimension of H. Then a set $\{\phi_m\}_{m=1}^M$ becomes a PONB in H if a set $\{\sqrt{k}\phi_m\}_{m=1}^M$ consists of k sets of ONBs in H.*

A proof of Theorem 5.25 is given in Section 5.8.12.

5.8 Proofs

In this section, proofs of all theorems, corollaries, and lemmas given in this chapter are provided.

5.8.1 Theorem 5.2

$A_{\mathcal{X}}^* A_{\mathcal{X}}$ is *positive definite* because of Eq.(5.15). Therefore, it has μ positive eigenvalues $\{\lambda_p\}_{p=1}^{\mu}$ considering the *geometric multiplicity*, where μ is the dimension of H. Then it holds that

$$\mathrm{tr} A_{\mathcal{X}}^* A_{\mathcal{X}} = \sum_{p=1}^{\mu} \lambda_p, \tag{5.130}$$

$$\mathrm{tr}(A_{\mathcal{X}}^* A_{\mathcal{X}})^{-1} = \sum_{p=1}^{\mu} \frac{1}{\lambda_p}. \tag{5.131}$$

It is well-known that the *arithmetic* and *harmonic means* have the following relation:

$$\frac{\sum_{p=1}^{\mu} \lambda_p}{\mu} \geq \frac{\mu}{\sum_{p=1}^{\mu} \frac{1}{\lambda_p}}, \tag{5.132}$$

where equality holds if and only if

$$\lambda_1 = \lambda_2 = \cdots = \lambda_\mu. \tag{5.133}$$

From Eqs.(5.23), (5.131), (5.132), and (5.130), we have

$$J_G[\mathcal{X}] = \sigma^2 \mathrm{tr}(A_{\mathcal{X}}^* A_{\mathcal{X}})^{-1} = \sigma^2 \sum_{p=1}^{\mu} \frac{1}{\lambda_p} \geq \frac{\sigma^2 \mu^2}{\sum_{p=1}^{\mu} \lambda_p} = \frac{\sigma^2 \mu^2}{\mathrm{tr} A_{\mathcal{X}}^* A_{\mathcal{X}}}. \tag{5.134}$$

It follows from Eqs.(5.9), (5.13), and (5.7) that

$$\begin{aligned} \mathrm{tr} A_{\mathcal{X}}^* A_{\mathcal{X}} &= \mathrm{tr}\left(\sum_{m=1}^{M} \left(K(\cdot, \boldsymbol{x}_m) \otimes \overline{K(\cdot, \boldsymbol{x}_m)} \right) \right) = \sum_{m=1}^{M} \|K(\cdot, \boldsymbol{x}_m)\|^2 \\ &= \sum_{m=1}^{M} \langle K(\cdot, \boldsymbol{x}_m), K(\cdot, \boldsymbol{x}_m) \rangle = \sum_{m=1}^{M} K(\boldsymbol{x}_m, \boldsymbol{x}_m) = \sum_{m=1}^{M} r \\ &= rM. \end{aligned} \tag{5.135}$$

Therefore, Eq.(5.134) yields

$$J_G[\mathcal{X}] \geq \frac{\sigma^2 \mu^2}{rM}. \tag{5.136}$$

From Eqs.(5.130), (5.135), and (5.133), equality in Eq.(5.136) holds if and only if

$$\lambda_1 = \lambda_2 = \cdots = \lambda_\mu = \frac{rM}{\mu}. \tag{5.137}$$

Since $\mathcal{R}(A_{\mathcal{X}}^*) = H$ as assumed in Eq.(5.15), Eq.(5.137) is equivalent to

$$A_{\mathcal{X}}^* A_{\mathcal{X}} = \frac{rM}{\mu} I_H, \tag{5.138}$$

which implies Eq.(5.24). Eq.(5.25) is clear from Eq.(5.136) with equality. ■

5.8.2 Lemma 5.3

If we let $\varphi'_m = \boldsymbol{e}_m$ and $\phi_m = \sqrt{\frac{\mu}{rM}} K(\cdot, \boldsymbol{x}_m)$ in Y defined by Eq.(5.121), then Eq.(5.26) is clear from Items 2 and 3 in Proposition 5.22. For any $\boldsymbol{v}$ in $\mathbf{C}^M$, it follows from Eq.(5.24) that

$$\begin{aligned}
\|A_{\mathcal{X}}^\dagger \boldsymbol{v}\| &= \sqrt{\|A_{\mathcal{X}}^\dagger \boldsymbol{v}\|^2} = \sqrt{\langle (A_{\mathcal{X}}^\dagger)^* A_{\mathcal{X}}^\dagger \boldsymbol{v}, \boldsymbol{v}\rangle} = \sqrt{\langle (A_{\mathcal{X}}^\dagger)^* (A_{\mathcal{X}}^* A_{\mathcal{X}})^{-1} A_{\mathcal{X}}^* \boldsymbol{v}, \boldsymbol{v}\rangle} \\
&= \sqrt{\langle (A_{\mathcal{X}}^\dagger)^* (\frac{rM}{\mu} I_H)^{-1} A_{\mathcal{X}}^* \boldsymbol{v}, \boldsymbol{v}\rangle} = \sqrt{\frac{\mu}{rM} \langle (A_{\mathcal{X}}^*)^\dagger A_{\mathcal{X}}^* \boldsymbol{v}, \boldsymbol{v}\rangle} \\
&= \sqrt{\frac{\mu}{rM} \langle P_{\mathcal{R}(A_{\mathcal{X}})} \boldsymbol{v}, \boldsymbol{v}\rangle} = \sqrt{\frac{\mu}{rM} \|P_{\mathcal{R}(A_{\mathcal{X}})} \boldsymbol{v}\|^2} \\
&= \sqrt{\frac{\mu}{rM}} \|P_{\mathcal{R}(A_{\mathcal{X}})} \boldsymbol{v}\|,
\end{aligned} \tag{5.139}$$

where $P_{\mathcal{R}(A_{\mathcal{X}})}$ denotes the orthogonal projection operator onto the range of $A_{\mathcal{X}}$. Eq.(5.139) implies Eq.(5.27). ■

5.8.3 Theorem 5.5

Let W be an operator from $\mathbf{C}^\mu$ to H defined as

$$W = \sum_{p=1}^{\mu} (\varphi_p \otimes \overline{\boldsymbol{e}_p}), \tag{5.140}$$

where $\boldsymbol{e}_p$ is the p-th vector of the so-called standard basis in $\mathbf{C}^\mu$. Since $\{\varphi_p(\boldsymbol{x})\}_{p=1}^\mu$ is an orthonormal basis in H, the operator W is *unitary*, i.e., it holds that

$$W^* = W^{-1}. \tag{5.141}$$

Then it follows from Eqs.(5.9), (5.140), and (5.13) that

$$\begin{aligned} A_{\mathcal{X}}W &= \sum_{m=1}^{M} \left(\boldsymbol{e}_m \otimes \overline{K(\cdot, \boldsymbol{x}_m)} \right) \sum_{p=1}^{\mu} (\varphi_p \otimes \overline{\boldsymbol{e}_p}) \\ &= \sum_{m=1}^{M} \langle \varphi_p, K(\cdot, \boldsymbol{x}_m) \rangle \, (\boldsymbol{e}_m \otimes \overline{\boldsymbol{e}_p}) \\ &= \sum_{m=1}^{M} \varphi_p(\boldsymbol{x}_m) \, (\boldsymbol{e}_m \otimes \overline{\boldsymbol{e}_p}) \,, \end{aligned} \tag{5.142}$$

which implies

$$[A_{\mathcal{X}}W]_{m,p} = \varphi_p(\boldsymbol{x}_m). \tag{5.143}$$

Therefore, $C_{\mathcal{X}}$ defined by Eq.(5.34) is expressed as

$$C_{\mathcal{X}} = W^* A_{\mathcal{X}}^* A_{\mathcal{X}} W. \tag{5.144}$$

When the sample points satisfy Condition (5.24), it follows from Eqs.(5.144) and (5.141) that

$$C_{\mathcal{X}} = W^*(\frac{rM}{\mu} I_H)W = \frac{rM}{\mu} W^{-1}W = \frac{rM}{\mu} I_\mu, \tag{5.145}$$

where I_μ is the μ-dimensional identity matrix. Substituting Eq.(5.145) into Eq.(5.33), we have Eq.(5.37). ■

5.8.4 Theorem 5.8

Any function $f(\boldsymbol{x})$ in a trigonometric polynomial space of order $(N_1, N_2, \ldots, N_L)$ can be expressed as

$$f(\boldsymbol{x}) = \sum_{n_1=-N_1}^{N_1} \sum_{n_2=-N_2}^{N_2} \cdots \sum_{n_L=-N_L}^{N_L} a_{n_1,n_2,\ldots,n_L} \prod_{l=1}^{L} \exp\left(i n_l \xi^{(l)}\right), \tag{5.146}$$

where $a_{n_1,n_2,\ldots,n_L}$ is a coefficient. It follows from Eqs.(5.13), (5.146), and (5.49) that

$$\begin{aligned} &\sum_{m_1=1}^{M_1} \sum_{m_2=1}^{M_2} \cdots \sum_{m_L=1}^{M_L} \left| \langle f, \frac{1}{\sqrt{M}} K(\cdot, \boldsymbol{x}_m) \rangle \right|^2 \\ &\quad = \frac{1}{M} \sum_{m_1=1}^{M_1} \sum_{m_2=1}^{M_2} \cdots \sum_{m_L=1}^{M_L} |f(\boldsymbol{x}_m)|^2 \\ &\quad = \frac{1}{M} \sum_{m_1=1}^{M_1} \sum_{m_2=1}^{M_2} \cdots \sum_{m_L=1}^{M_L} \left| \sum_{n_1=-N_1}^{N_1} \sum_{n_2=-N_2}^{N_2} \cdots \sum_{n_L=-N_L}^{N_L} a_{n_1,n_2,\ldots,n_L} \prod_{l=1}^{L} \exp(i n_l \xi_m^{(l)}) \right|^2 \end{aligned}$$

$$= \frac{1}{M} \sum_{m_1=1}^{M_1} \sum_{m_2=1}^{M_2} \cdots \sum_{m_L=1}^{M_L} \sum_{n_1=-N_1}^{N_1} \sum_{n_2=-N_2}^{N_2} \cdots \sum_{n_L=-N_L}^{N_L}$$
$$\sum_{n_1'=-N_1}^{N_1} \sum_{n_2'=-N_2}^{N_2} \cdots \sum_{n_L'=-N_L}^{N_L} a_{n_1,n_2,\ldots,n_L} \overline{a_{n_1',n_2',\ldots,n_L'}} \prod_{l=1}^{L} \exp\left(i(n_l - n_l')\xi_m^{(l)}\right)$$
$$= \frac{1}{M} \sum_{n_1=-N_1}^{N_1} \sum_{n_2=-N_2}^{N_2} \cdots \sum_{n_L=-N_L}^{N_L} \sum_{n_1'=-N_1}^{N_1} \sum_{n_2'=-N_2}^{N_2} \cdots \sum_{n_L'=-N_L}^{N_L} a_{n_1,n_2,\ldots,n_L} \overline{a_{n_1',n_2',\ldots,n_L'}}$$
$$\times \prod_{l=1}^{L} \left[\sum_{m_l=1}^{M_l} \exp\left(i(n_l - n_l')\frac{2\pi m_l}{M_l}\right) \right] \prod_{l=1}^{L} \exp\left(i(n_l - n_l')(c_l - \frac{2\pi}{M_l})\right). \quad (5.147)$$

For any integers n_l and n_l', it generally holds that

$$\sum_{m_l=1}^{M_l} \exp\left(i(n_l - n_l')\frac{2\pi m_l}{M_l}\right) = \begin{cases} 0 & \text{if } n_l \neq n_l', \\ M_l & \text{if } n_l = n_l'. \end{cases} \quad (5.148)$$

Therefore, it follows from Eqs.(5.147), (5.148), (5.46), and (5.146) that

$$\sum_{m_1=1}^{M_1} \sum_{m_2=1}^{M_2} \cdots \sum_{m_L=1}^{M_L} \left| \langle f, \frac{1}{\sqrt{M}} K(\cdot, \boldsymbol{x}_m) \rangle \right|^2$$
$$= \frac{1}{M} \sum_{n_1=-N_1}^{N_1} \sum_{n_2=-N_2}^{N_2} \cdots \sum_{n_L=-N_L}^{N_L} |a_{n_1,n_2,\ldots,n_L}|^2 \prod_{l=1}^{L} M_l \prod_{l=1}^{L} \exp(0)$$
$$= \sum_{n_1=-N_1}^{N_1} \sum_{n_2=-N_2}^{N_2} \cdots \sum_{n_L=-N_L}^{N_L} |a_{n_1,n_2,\ldots,n_L}|^2$$
$$= \|f\|^2. \quad (5.149)$$

According to Items 1 and 2 in Proposition 5.21 with $\phi_m = \frac{1}{\sqrt{M}} K(\cdot, \boldsymbol{x}_m)$, Eq.(5.149) is equivalent to that a set $\{\frac{1}{\sqrt{M}} K(\cdot, \boldsymbol{x}_m)\}_{m=1}^{M}$ forms a POB in H. Therefore, Items 1 and 2 in Proposition 5.22 with $\varphi_m' = \boldsymbol{e}_m$ and $\phi_m = \frac{1}{\sqrt{M}} K(\cdot, \boldsymbol{x}_m)$ yields Eq.(5.24) with r given by Eq.(5.40). ■

5.8.5 Theorem 5.9

For a set $\{\boldsymbol{x}_m\}_{m=(t-1)\mu+1}^{t\mu}$ of μ sample points with a fixed t, it follows from Eqs.(3.29), (3.30), (5.51), and (5.52) that

$$\frac{1}{\mu} K(\boldsymbol{x}_m, \boldsymbol{x}_{m'}) = \delta_{mm'}, \quad (5.150)$$

where $\delta_{mm'}$ denotes *Kronecker's delta* defined as

$$\delta_{mm'} = \begin{cases} 0 & \text{if } m \neq m', \\ 1 & \text{if } m = m'. \end{cases} \tag{5.151}$$

Therefore, it follows from Eqs.(5.13) and (5.150) that

$$\begin{aligned} \langle \sqrt{\frac{k}{M}} K(\cdot, \boldsymbol{x}_{m'}), \sqrt{\frac{k}{M}} K(\cdot, \boldsymbol{x}_m) \rangle &= \frac{k}{M} K(\boldsymbol{x}_m, \boldsymbol{x}_{m'}) \\ &= \delta_{mm'}. \end{aligned} \tag{5.152}$$

Eq.(5.152) implies that for each t, a set $\{\sqrt{\frac{k}{M}} K(\cdot, \boldsymbol{x}_m)\}_{m=(t-1)\mu+1}^{t\mu}$ of μ elements in H forms an orthonormal basis in H. Therefore, a set $\{\frac{1}{\sqrt{M}} K(\cdot, \boldsymbol{x}_m)\}_{m=1}^{M}$ forms a PONB in H from Theorem 5.25. This is equivalent to Eq.(5.24) with r given by Eq.(5.40) according to Proposition 5.22 with $\varphi'_m = \boldsymbol{e}_m$ and $\phi_m = \frac{1}{\sqrt{M}} K(\cdot, \boldsymbol{x}_m)$. ■

5.8.6 Theorem 5.11

Let Γ_{m+1} be a matrix from $\mathbf{C}^m$ to $\mathbf{C}^{m+1}$ defined as

$$\Gamma_{m+1} = \sum_{j=1}^{m} \left(\boldsymbol{e}_j^{(m+1)} \otimes \overline{\boldsymbol{e}_j^{(m)}} \right). \tag{5.153}$$

It follows from Eqs.(5.61) and (5.153) that the operator A_{m+1} is expressed by using A_m as

$$\begin{aligned} A_{m+1} &= \sum_{j=1}^{m+1} \left(\boldsymbol{e}_j^{(m)} \otimes \overline{K(\cdot, \boldsymbol{x}_j)} \right) \\ &= \Gamma_{m+1} A_m + \boldsymbol{e}_{m+1}^{(m+1)} \otimes \overline{K(\cdot, \boldsymbol{x}_{m+1})}. \end{aligned} \tag{5.154}$$

From Theorem 4.3 in Albert [5], we have

$$A_{m+1}^{\dagger} = A_m^{\dagger} \Gamma_{m+1}^{*} + g_{m+1} \otimes \overline{\left(\boldsymbol{e}_{m+1}^{(m+1)} - \Gamma_{m+1} (A_m^{\dagger})^{*} K(\cdot, \boldsymbol{x}_{m+1}) \right)}, \tag{5.155}$$

where g_{m+1} is a function in H defined as

$$g_{m+1} = \begin{cases} \dfrac{P_{\mathcal{N}(A_m)} K(\cdot, \boldsymbol{x}_{m+1})}{\| P_{\mathcal{N}(A_m)} K(\cdot, \boldsymbol{x}_{m+1}) \|^2} & \text{if } K(\boldsymbol{x}, \boldsymbol{x}_{m+1}) \notin \mathcal{L}\left(\{K(\boldsymbol{x}, \boldsymbol{x}_j)\}_{j=1}^{m} \right), \\ \dfrac{(A_m^{*} A_m)^{\dagger} K(\cdot, \boldsymbol{x}_{m+1})}{1 + \langle (A_m^{*} A_m)^{\dagger} K(\cdot, \boldsymbol{x}_{m+1}), K(\cdot, \boldsymbol{x}_{m+1}) \rangle} & \\ \qquad \text{if } K(\boldsymbol{x}, \boldsymbol{x}_{m+1}) \in \mathcal{L}\left(\{K(\boldsymbol{x}, \boldsymbol{x}_j)\}_{j=1}^{m} \right). \end{cases} \tag{5.156}$$

Then it follows from Eqs.(5.67), (5.155), (5.153), (5.60), and (5.13) that

$$\begin{aligned}
\hat{f}_{m+1} &= A_{m+1}^{\dagger}\boldsymbol{y}^{(m+1)} \\
&= \left(A_m^{\dagger}\Gamma_{m+1}^{*} + g_{m+1}\otimes\overline{\left(\boldsymbol{e}_{m+1}^{(m+1)} - \Gamma_{m+1}(A_m^{\dagger})^{*}K(\cdot,\boldsymbol{x}_{m+1})\right)}\right)\boldsymbol{y}^{(m+1)} \\
&= A_m^{\dagger}\Gamma_{m+1}^{*}\boldsymbol{y}^{(m+1)} + \left(\langle \boldsymbol{y}^{(m+1)}, \boldsymbol{e}_{m+1}^{(m+1)}\rangle - \langle A_m^{\dagger}\Gamma_{m+1}^{*}\boldsymbol{y}^{(m+1)}, K(\cdot,\boldsymbol{x}_{m+1})\rangle\right) g_{m+1} \\
&= A_m^{\dagger}\boldsymbol{y}^{(m)} + \left(y_{m+1} - \langle A_m^{\dagger}\boldsymbol{y}^{(m)}, K(\cdot,\boldsymbol{x}_{m+1})\rangle\right) g_{m+1} \\
&= \hat{f}_m + \left(y_{m+1} - \langle \hat{f}_m, K(\cdot,\boldsymbol{x}_{m+1})\rangle\right) g_{m+1} \\
&= \hat{f}_m + \left(y_{m+1} - \hat{f}_m(\boldsymbol{x}_{m+1})\right) g_{m+1}. \qquad (5.157)
\end{aligned}$$

Eqs.(5.157) and (5.156) imply Eqs.(5.68) and (5.69). ■

5.8.7 Lemma 5.12

First, we give a proof for $J_b^{(m+1)}$. It follows from Eqs.(5.67), (5.63), and (5.65) that

$$\begin{aligned}
\|\mathrm{E}_{\boldsymbol{\epsilon}}\hat{f}_{m+1} - f\|^2 &= \|\mathrm{E}_{\boldsymbol{\epsilon}}A_{m+1}^{\dagger}\boldsymbol{y}^{(m+1)} - f\|^2 \\
&= \|\mathrm{E}_{\boldsymbol{\epsilon}}A_{m+1}^{\dagger}(A_{m+1}f + \boldsymbol{\epsilon}^{(m+1)}) - f\|^2 \\
&= \|A_{m+1}^{\dagger}A_{m+1}f - f\|^2 \\
&= \|P_{\mathcal{R}(A_{m+1}^{*})}f - f\|^2, \qquad (5.158)
\end{aligned}$$

where $P_{\mathcal{R}(A_{m+1}^{*})}$ is the orthogonal projection operator onto the range of A_{m+1}^{*}. When $K(\boldsymbol{x},\boldsymbol{x}_{m+1}) \notin \mathcal{L}\left(\{K(\boldsymbol{x},\boldsymbol{x}_j)\}_{j=1}^{m}\right)$, it holds that

$$P_{\mathcal{R}(A_{m+1}^{*})} = P_{\mathcal{R}(A_m^{*})} + P_{m+1}. \qquad (5.159)$$

Here, P_{m+1} is the orthogonal projection operator onto a one-dimensional subspace spanned by $P_{\mathcal{N}(A_m)}K(\cdot,\boldsymbol{x}_{m+1})$:

$$P_{m+1} = \frac{P_{\mathcal{N}(A_m)}K(\cdot,\boldsymbol{x}_{m+1}) \otimes \overline{P_{\mathcal{N}(A_m)}K(\cdot,\boldsymbol{x}_{m+1})}}{\|P_{\mathcal{N}(A_m)}K(\cdot,\boldsymbol{x}_{m+1})\|^2}, \qquad (5.160)$$

where $P_{\mathcal{N}(A_m)}$ is the orthogonal projection operator onto the null space of A_m. Therefore, it follows from Eqs.(5.158) and (5.159) that

$$\begin{aligned}
\|\mathrm{E}_{\boldsymbol{\epsilon}}\hat{f}_{m+1} - f\|^2 &= \|(P_{\mathcal{R}(A_m^{*})} + P_{m+1})f - f\|^2 \\
&= \|P_{\mathcal{R}(A_m^{*})}f - f\|^2 - 2\mathrm{Re}\langle P_{\mathcal{R}(A_m^{*})}f - f, P_{m+1}f\rangle + \|P_{m+1}f\|^2, \qquad (5.161)
\end{aligned}$$

where 'Re' denotes the real part of a complex number. Since $P_{m+1}P_{\mathcal{N}(A_m)} = 0$, it follows from Eqs.(5.161) and (5.160) that

$$\begin{aligned}\|\mathrm{E}_{\boldsymbol{\epsilon}}\hat{f}_{m+1} - f\|^2 &= \|P_{\mathcal{R}(A_m^*)}f - f\|^2 - 2\langle P_{m+1}f, f\rangle + \|P_{m+1}f\|^2 \\ &= \|P_{\mathcal{R}(A_m^*)}f - f\|^2 - \|P_{m+1}f\|^2 \\ &= \|\mathrm{E}_{\boldsymbol{\epsilon}}\hat{f}_m - f\|^2 \\ &\quad - \left\| \frac{\langle f, P_{\mathcal{N}(A_m)}K(\cdot, \boldsymbol{x}_{m+1})\rangle}{\|P_{\mathcal{N}(A_m)}K(\cdot, \boldsymbol{x}_{m+1})\|^2} P_{\mathcal{N}(A_m)}K(\cdot, \boldsymbol{x}_{m+1}) \right\|^2, \end{aligned} \tag{5.162}$$

which implies Eq.(5.72). When $K(\boldsymbol{x}, \boldsymbol{x}_{m+1}) \in \mathcal{L}\left(\{K(\boldsymbol{x}, \boldsymbol{x}_j)\}_{j=1}^m\right)$, it holds that

$$P_{\mathcal{R}(A_{m+1}^*)} = P_{\mathcal{R}(A_m^*)}. \tag{5.163}$$

Therefore, it follows from Eqs.(5.158) and (5.163) that

$$\begin{aligned}\|\mathrm{E}_{\boldsymbol{\epsilon}}\hat{f}_{m+1} - f\|^2 &= \|P_{\mathcal{R}(A_m^*)}f - f\|^2 \\ &= \|\mathrm{E}_{\boldsymbol{\epsilon}}\hat{f}_m - f\|^2, \end{aligned} \tag{5.164}$$

which implies Eq.(5.74).

Now we give a proof for $J_v^{(m+1)}$. It follows from Eqs.(5.67), (5.63), (5.65), and (5.66) that

$$\begin{aligned}\mathrm{E}_{\boldsymbol{\epsilon}}\|\hat{f}_{m+1} - \mathrm{E}_{\boldsymbol{\epsilon}}\hat{f}_{m+1}\|^2 &= \mathrm{E}_{\boldsymbol{\epsilon}}\|A_{m+1}^\dagger \boldsymbol{y}^{(m+1)} - \mathrm{E}_{\boldsymbol{\epsilon}}A_{m+1}^\dagger \boldsymbol{y}^{(m+1)}\|^2 \\ &= \mathrm{E}_{\boldsymbol{\epsilon}}\|A_{m+1}^\dagger(A_{m+1}f + \boldsymbol{\epsilon}^{(m+1)}) - \mathrm{E}_{\boldsymbol{\epsilon}}A_{m+1}^\dagger(A_{m+1}f + \boldsymbol{\epsilon}^{(m+1)})\|^2 \\ &= \mathrm{E}_{\boldsymbol{\epsilon}}\|A_{m+1}^\dagger \boldsymbol{\epsilon}^{(m+1)}\|^2 \\ &= \mathrm{tr}\left(A_{m+1}^\dagger \mathrm{E}_{\boldsymbol{\epsilon}}\left(\boldsymbol{\epsilon}^{(m+1)} \otimes \overline{\boldsymbol{\epsilon}^{(m+1)}}\right)(A_{m+1}^\dagger)^*\right) \\ &= \sigma^2 \mathrm{tr} A_{m+1}^\dagger (A_{m+1}^\dagger)^*. \end{aligned} \tag{5.165}$$

By using the fact that

$$\Gamma_{m+1}^* \Gamma_{m+1} = I_m, \tag{5.166}$$

$$\Gamma_{m+1}^* \boldsymbol{e}_{m+1}^{(m+1)} = 0, \tag{5.167}$$

it follows from Eq.(5.155) that

$$\begin{aligned}A_{m+1}^\dagger (A_{m+1}^\dagger)^* = &\left(A_m^\dagger \Gamma_{m+1}^* + g_{m+1} \otimes \overline{\left(\boldsymbol{e}_{m+1}^{(m+1)} - \Gamma_{m+1}(A_m^\dagger)^* K(\cdot, \boldsymbol{x}_{m+1})\right)}\right) \\ &\times \left(\Gamma_{m+1}(A_m^\dagger)^* + \left(\boldsymbol{e}_{m+1}^{(m+1)} - \Gamma_{m+1}(A_m^\dagger)^* K(\cdot, \boldsymbol{x}_{m+1})\right) \otimes \overline{g_{m+1}}\right) \end{aligned}$$

$$= A_m^\dagger (A_m^\dagger)^* - A_m^\dagger (A_m^\dagger)^* K(\cdot, \boldsymbol{x}_{m+1}) \otimes \overline{g_{m+1}}$$
$$- g_{m+1} \otimes \overline{A_m^\dagger (A_m^\dagger)^* K(\cdot, \boldsymbol{x}_{m+1})}$$
$$+ (1 + \langle A_m^\dagger (A_m^\dagger)^* K(\cdot, \boldsymbol{x}_{m+1}), K(\cdot, \boldsymbol{x}_{m+1}) \rangle) (g_{m+1} \otimes \overline{g_{m+1}}) . \quad (5.168)$$

From Eqs.(5.165) and (5.168), we have

$$\mathrm{E}_\epsilon \| \hat{f}_{m+1} - \mathrm{E}_\epsilon \hat{f}_{m+1} \|^2 = \mathrm{E}_\epsilon \| \hat{f}_m - \mathrm{E}_\epsilon \hat{f}_m \|^2$$
$$- 2\sigma^2 \mathrm{Re} \langle (A_m^* A_m)^\dagger K(\cdot, \boldsymbol{x}_{m+1}), g_{m+1} \rangle$$
$$+ \sigma^2 (1 + \langle (A_m^* A_m)^\dagger K(\cdot, \boldsymbol{x}_{m+1}), K(\cdot, \boldsymbol{x}_{m+1}) \rangle) \| g_{m+1} \|^2 . \quad (5.169)$$

When $K(\boldsymbol{x}, \boldsymbol{x}_{m+1}) \notin \mathcal{L}\left(\{K(\boldsymbol{x}, \boldsymbol{x}_j)\}_{j=1}^m\right)$, it follows from Eqs.(5.169) and (5.156) that

$$\mathrm{E}_\epsilon \| \hat{f}_{m+1} - \mathrm{E}_\epsilon \hat{f}_{m+1} \|^2 = \mathrm{E}_\epsilon \| \hat{f}_m - \mathrm{E}_\epsilon \hat{f}_m \|^2$$
$$+ \sigma^2 \frac{1 + \langle (A_m^* A_m)^\dagger K(\cdot, \boldsymbol{x}_{m+1}), K(\cdot, \boldsymbol{x}_{m+1}) \rangle}{\| P_{\mathcal{N}(A_m)} K(\cdot, \boldsymbol{x}_{m+1}) \|^2}, \quad (5.170)$$

which implies Eq.(5.73). When $K(\boldsymbol{x}, \boldsymbol{x}_{m+1}) \in \mathcal{L}\left(\{K(\boldsymbol{x}, \boldsymbol{x}_j)\}_{j=1}^m\right)$, it follows from Eqs.(5.169) and (5.156) that

$$\mathrm{E}_\epsilon \| \hat{f}_{m+1} - \mathrm{E}_\epsilon \hat{f}_{m+1} \|^2 = \mathrm{E}_\epsilon \| \hat{f}_m - \mathrm{E}_\epsilon \hat{f}_m \|^2$$
$$- 2\sigma^2 \frac{\| (A_m^* A_m)^\dagger K(\cdot, \boldsymbol{x}_{m+1}) \|^2}{1 + \langle (A_m^* A_m)^\dagger K(\cdot, \boldsymbol{x}_{m+1}), K(\cdot, \boldsymbol{x}_{m+1}) \rangle}$$
$$+ \sigma^2 \frac{\| (A_m^* A_m)^\dagger K(\cdot, \boldsymbol{x}_{m+1}) \|^2}{1 + \langle (A_m^* A_m)^\dagger K(\cdot, \boldsymbol{x}_{m+1}), K(\cdot, \boldsymbol{x}_{m+1}) \rangle}, \quad (5.171)$$

which implies Eq.(5.75). ■

5.8.8 Lemma 5.14

It follows from Eq.(5.158) that the bias of $\hat{f}_m(\boldsymbol{x})$ yields

$$\| \mathrm{E}_\epsilon \hat{f}_m - f \|^2 = \| P_{\mathcal{R}(A_m^*)} f - f \|^2, \quad (5.172)$$

which vanishes if $\mathcal{L}\left(\{K(\boldsymbol{x}, \boldsymbol{x}_j)\}_{j=1}^m\right) = H$. ■

5.8.9 Corollary 5.15

Let W be an operator from $\mathbf{C}^\mu$ to H defined by Eq.(5.140). Since $\{\varphi_p(\boldsymbol{x})\}_{p=1}^\mu$ is an orthonormal basis in H, the operator W is *unitary* and Eq.(5.141) holds. It follows from

Eqs.(5.61), (5.140), (5.13), and (5.79) that

$$\begin{aligned} A_m W &= \sum_{j=1}^{m} \left(\boldsymbol{e}_j \otimes \overline{K(\cdot, \boldsymbol{x}_j)} \right) \sum_{p=1}^{\mu} (\varphi_p \otimes \overline{\boldsymbol{e}_p}) \\ &= \sum_{j=1}^{m} \langle \varphi_p, K(\cdot, \boldsymbol{x}_j) \rangle \, (\boldsymbol{e}_j \otimes \overline{\boldsymbol{e}_p}) \\ &= \sum_{j=1}^{m} \varphi_p(\boldsymbol{x}_j) \, (\boldsymbol{e}_j \otimes \overline{\boldsymbol{e}_p}) \\ &= B_m. \end{aligned} \tag{5.173}$$

Operating W^{-1} from the right-hand side of Eq.(5.173) and using Eq.(5.141), we have

$$A_m = B_m W^*. \tag{5.174}$$

In this case, it follows from Theorem 4.11 in Albert [5] that

$$A_m^\dagger = W B_m^\dagger. \tag{5.175}$$

From Eqs.(5.175) and (5.80), we have

$$\begin{aligned} (A_m^* A_m)^\dagger &= A_m^\dagger (A_m^\dagger)^* = W B_m^\dagger (W B_m^\dagger)^* \\ &= W B_m^\dagger (B_m^\dagger)^* W^* = W (B_m^* B_m)^\dagger W^* \\ &= W C_m^\dagger W^*. \end{aligned} \tag{5.176}$$

It follows from Eqs.(5.141), (5.175), (5.174), and (5.81) that $P_{\mathcal{N}(A_m)}$, the orthogonal projection operator onto the null space of A_m, is expressed as

$$\begin{aligned} P_{\mathcal{N}(A_m)} &= I_H - A_m^\dagger A_m = W W^* - W B_m^\dagger B_m W^* = W (I_\mu - B_m^\dagger B_m) W^* \\ &= W G_m W^*. \end{aligned} \tag{5.177}$$

From Eqs.(5.140), (5.13), and (5.82), we have

$$\begin{aligned} W^* K(\cdot, \boldsymbol{x}_{m+1}) &= \sum_{p=1}^{\mu} (\boldsymbol{e}_p \otimes \overline{\varphi_p}) \, K(\cdot, \boldsymbol{x}_{m+1}) = \sum_{p=1}^{\mu} \langle K(\cdot, \boldsymbol{x}_{m+1}), \varphi_p \rangle \boldsymbol{e}_p \\ &= \sum_{p=1}^{\mu} \overline{\langle \varphi_p, K(\cdot, \boldsymbol{x}_{m+1}) \rangle} \boldsymbol{e}_p = \sum_{p=1}^{\mu} \overline{\varphi_p(\boldsymbol{x}_{m+1})} \boldsymbol{e}_p \\ &= d_{m+1}. \end{aligned} \tag{5.178}$$

Then it follows from Eqs.(5.73), (5.176), (5.177), and (5.178) that

$$\begin{aligned} J_v^{(m+1)} &= \sigma^2 \frac{1 + \langle (A_m^* A_m)^\dagger K(\cdot, \boldsymbol{x}_{m+1}), K(\cdot, \boldsymbol{x}_{m+1}) \rangle}{\langle P_{\mathcal{N}(A_m)} K(\cdot, \boldsymbol{x}_{m+1}), K(\cdot, \boldsymbol{x}_{m+1}) \rangle} \\ &= \sigma^2 \frac{1 + \langle W C_m^\dagger W^* K(\cdot, \boldsymbol{x}_{m+1}), K(\cdot, \boldsymbol{x}_{m+1}) \rangle}{\langle W G_m W^* K(\cdot, \boldsymbol{x}_{m+1}), K(\cdot, \boldsymbol{x}_{m+1}) \rangle} \\ &= \sigma^2 \frac{1 + \langle C_m^\dagger d_{m+1}, d_{m+1} \rangle}{\langle G_m d_{m+1}, d_{m+1} \rangle}, \end{aligned} \tag{5.179}$$

which implies Eq.(5.83). Similarly, it follows from Eqs.(5.75), (5.176), and (5.178) that

$$\begin{aligned} J_v^{(m+1)} &= -\sigma^2 \frac{\| (A_m^* A_m)^\dagger K(\cdot, \boldsymbol{x}_{m+1}) \|^2}{1 + \langle (A_m^* A_m)^\dagger K(\cdot, \boldsymbol{x}_{m+1}), K(\cdot, \boldsymbol{x}_{m+1}) \rangle} \\ &= -\sigma^2 \frac{\| W C_m^\dagger W^* K(\cdot, \boldsymbol{x}_{m+1}) \|^2}{1 + \langle W C_m^\dagger W^* K(\cdot, \boldsymbol{x}_{m+1}), K(\cdot, \boldsymbol{x}_{m+1}) \rangle} \\ &= -\sigma^2 \frac{\| C_m^\dagger d_{m+1} \|^2}{1 + \langle C_m^\dagger d_{m+1}, d_{m+1} \rangle}, \end{aligned} \tag{5.180}$$

which implies Eq.(5.84). ■

5.8.10 Corollary 5.16

It follows from Eqs.(5.140) and (5.86) that

$$\begin{aligned} W \boldsymbol{w}^{(m+1)} &= \sum_{p=1}^{\mu} (\varphi_p \otimes \overline{e_p}) \, \boldsymbol{w}^{(m+1)} = \sum_{p=1}^{\mu} [\boldsymbol{w}^{(m+1)}]_p \varphi_p \\ &= \hat{f}_{m+1}. \end{aligned} \tag{5.181}$$

When $K(\boldsymbol{x}, \boldsymbol{x}_{m+1}) \notin \mathcal{L}\left(\{K(\boldsymbol{x}, \boldsymbol{x}_j)\}_{j=1}^m\right)$, it follows from Eqs.(5.181), (5.68), (5.13), (5.177), and (5.178) that

$$\begin{aligned} W \boldsymbol{w}^{(m+1)} &= \hat{f}_m(\boldsymbol{x}) + \frac{y_{m+1} - \langle \hat{f}_m, K(\cdot, \boldsymbol{x}_{m+1}) \rangle}{\langle P_{\mathcal{N}(A_m)} K(\cdot, \boldsymbol{x}_{m+1}), K(\cdot, \boldsymbol{x}_{m+1}) \rangle} P_{\mathcal{N}(A_m)} K(\boldsymbol{x}, \boldsymbol{x}_{m+1}) \\ &= W \boldsymbol{w}^{(m)} + \frac{y_{m+1} - \langle W \boldsymbol{w}^{(m)}, K(\cdot, \boldsymbol{x}_{m+1}) \rangle}{\langle W G_m W^* K(\cdot, \boldsymbol{x}_{m+1}), K(\cdot, \boldsymbol{x}_{m+1}) \rangle} W G_m W^* K(\boldsymbol{x}, \boldsymbol{x}_{m+1}) \\ &= W \boldsymbol{w}^{(m)} + \frac{y_{m+1} - \langle \boldsymbol{w}^{(m)}, d_{m+1} \rangle}{\langle G_m d_{m+1}, d_{m+1} \rangle} W G_m d_{m+1}. \end{aligned} \tag{5.182}$$

Operating W^{-1} from the left-hand side of Eq.(5.182), we have Eq.(5.87). When $K(\boldsymbol{x}, \boldsymbol{x}_{m+1}) \in \mathcal{L}\left(\{K(\boldsymbol{x}, \boldsymbol{x}_j)\}_{j=1}^m\right)$, if follows from Eqs.(5.181), (5.69), (5.13), (5.176), and

(5.178) that

$$
\begin{aligned}
W\boldsymbol{w}^{(m+1)} &= \hat{f}_m(\boldsymbol{x}) + \frac{y_{m+1} - \langle \hat{f}_m, K(\cdot, \boldsymbol{x}_{m+1})\rangle}{1 + \langle (A_m^* A_m)^\dagger K(\cdot, \boldsymbol{x}_{m+1}), K(\cdot, \boldsymbol{x}_{m+1})\rangle} (A_m^* A_m)^\dagger K(\boldsymbol{x}, \boldsymbol{x}_{m+1}) \\
&= W\boldsymbol{w}^{(m)} + \frac{y_{m+1} - \langle W\boldsymbol{w}^{(m)}, K(\cdot, \boldsymbol{x}_{m+1})\rangle}{1 + \langle WC_m^\dagger W^* K(\cdot, \boldsymbol{x}_{m+1}), K(\cdot, \boldsymbol{x}_{m+1})\rangle} WC_m^\dagger W^* K(\boldsymbol{x}, \boldsymbol{x}_{m+1}) \\
&= W\boldsymbol{w}^{(m)} + \frac{y_{m+1} - \langle \boldsymbol{w}^{(m)}, d_{m+1}\rangle}{1 + \langle C_m^\dagger d_{m+1}, d_{m+1}\rangle} WC_m^\dagger d_{m+1}. \qquad (5.183)
\end{aligned}
$$

Operating W^{-1} from the left-hand side of Eq.(5.183), we have Eq.(5.88). ■

5.8.11 Lemma 5.17

First, we give a proof for $C_{m+1}^\dagger$. Operating W^{-1} from the left-hand side of Eq.(5.175) and using Eq.(5.141), we have

$$
B_m^\dagger = W^* A_m^\dagger. \qquad (5.184)
$$

It follows from Eqs.(5.80) and (5.184) that

$$
\begin{aligned}
C_{m+1}^\dagger &= (B_{m+1}^* B_{m+1})^\dagger = B_{m+1}^\dagger (B_{m+1}^\dagger)^* = W^* A_{m+1}^\dagger (W^* A_{m+1}^\dagger)^* \\
&= W^* A_{m+1}^\dagger (A_{m+1}^\dagger)^* W. \qquad (5.185)
\end{aligned}
$$

Then it follows from Eqs.(5.185), (5.168), (5.141), and (5.178) that

$$
\begin{aligned}
C_{m+1}^\dagger = W^* \Big(& A_m^\dagger (A_m^\dagger)^* - A_m^\dagger (A_m^\dagger)^* W W^* K(\cdot, \boldsymbol{x}_{m+1}) \otimes \overline{g_{m+1}} \\
& - g_{m+1} \otimes \overline{A_m^\dagger (A_m^\dagger)^* W W^* K(\cdot, \boldsymbol{x}_{m+1})} \\
& + \Big(1 + \langle WW^* A_m^\dagger (A_m^\dagger)^* W W^* K(\cdot, \boldsymbol{x}_{m+1}), K(\cdot, \boldsymbol{x}_{m+1})\rangle\Big) \\
& \times (g_{m+1} \otimes \overline{g_{m+1}}) \Big) W \\
= C_m^\dagger - C_m^\dagger d_{m+1} \otimes \overline{W^* g_{m+1}} & - W^* g_{m+1} \otimes \overline{C_m^\dagger d_{m+1}} \\
& + (1 + \langle C_m^\dagger d_{m+1}, d_{m+1}\rangle) \left(W^* g_{m+1} \otimes \overline{W^* g_{m+1}}\right). \qquad (5.186)
\end{aligned}
$$

When $K(\boldsymbol{x}, \boldsymbol{x}_{m+1}) \notin \mathcal{L}\left(\{K(\boldsymbol{x}, \boldsymbol{x}_j)\}_{j=1}^m\right)$, it follows from Eqs.(5.156), (5.177), (5.141), and (5.178) that

$$
W^* g_{m+1} = \frac{W^* P_{\mathcal{N}(A_m)} K(\cdot, \boldsymbol{x}_{m+1})}{\langle P_{\mathcal{N}(A_m)} K(\cdot, \boldsymbol{x}_{m+1}), K(\cdot, \boldsymbol{x}_{m+1})\rangle}
$$

$$= \frac{W^*WG_mW^*K(\cdot, \boldsymbol{x}_{m+1})}{\langle WG_mW^*K(\cdot, \boldsymbol{x}_{m+1}), K(\cdot, \boldsymbol{x}_{m+1})\rangle}$$
$$= \frac{G_m d_{m+1}}{\langle G_m d_{m+1}, d_{m+1}\rangle}. \tag{5.187}$$

Eqs.(5.186) and (5.187) imply Eq.(5.89). When $K(\boldsymbol{x}, \boldsymbol{x}_{m+1}) \in \mathcal{L}\left(\{K(\boldsymbol{x}, \boldsymbol{x}_j)\}_{j=1}^m\right)$, it follows from Eqs.(5.156), (5.176), (5.141), and (5.178) that

$$W^*g_{m+1} = \frac{W^*(A_m^*A_m)^\dagger K(\cdot, \boldsymbol{x}_{m+1})}{1 + \langle (A_m^*A_m)^\dagger K(\cdot, \boldsymbol{x}_{m+1}), K(\cdot, \boldsymbol{x}_{m+1})\rangle}$$
$$= \frac{W^*WC_m^\dagger W^*K(\cdot, \boldsymbol{x}_{m+1})}{1 + \langle WC_m^\dagger W^*K(\cdot, \boldsymbol{x}_{m+1}), K(\cdot, \boldsymbol{x}_{m+1})\rangle}$$
$$= \frac{C_m^\dagger d_{m+1}}{1 + \langle C_m^\dagger d_{m+1}, d_{m+1}\rangle}. \tag{5.188}$$

Eqs.(5.186) and (5.188) imply Eq.(5.91).

Now we give a proof for G_{m+1}. It follows from Eqs.(5.81), (5.141), (5.184), and (5.173) that

$$G_{m+1} = I_\mu - B_{m+1}^\dagger B_{m+1} = W^*W - W^*A_{m+1}^\dagger A_{m+1}W$$
$$= W^*(I_\mu - P_{\mathcal{R}(A_{m+1}^*)})W. \tag{5.189}$$

When $K(\boldsymbol{x}, \boldsymbol{x}_{m+1}) \notin \mathcal{L}\left(\{K(\boldsymbol{x}, \boldsymbol{x}_j)\}_{j=1}^m\right)$, it follows from Eqs.(5.189), (5.159), (5.160), (5.177), and (5.178) that

$$G_{m+1} = W^*\left(I_\mu - P_{\mathcal{R}(A_m^*)} - P_{m+1}\right)W$$
$$= W^*\left(I_\mu - P_{\mathcal{R}(A_m^*)} - \frac{P_{\mathcal{N}(A_m)}K(\cdot, \boldsymbol{x}_{m+1}) \otimes \overline{P_{\mathcal{N}(A_m)}K(\cdot, \boldsymbol{x}_{m+1})}}{\|P_{\mathcal{N}(A_m)}K(\cdot, \boldsymbol{x}_{m+1})\|^2}\right)W$$
$$= W^*(I_\mu - A_m^\dagger A_m)W - W^*\left(\frac{P_{\mathcal{N}(A_m)}K(\cdot, \boldsymbol{x}_{m+1}) \otimes \overline{P_{\mathcal{N}(A_m)}K(\cdot, \boldsymbol{x}_{m+1})}}{\langle P_{\mathcal{N}(A_m)}K(\cdot, \boldsymbol{x}_{m+1}), K(\cdot, \boldsymbol{x}_{m+1})\rangle}\right)W$$
$$= (I_\mu - B_m^\dagger B_m) - W^*\left(\frac{WG_mW^*K(\cdot, \boldsymbol{x}_{m+1}) \otimes \overline{WG_mW^*K(\cdot, \boldsymbol{x}_{m+1})}}{\langle WG_mW^*K(\cdot, \boldsymbol{x}_{m+1}), K(\cdot, \boldsymbol{x}_{m+1})\rangle}\right)W$$
$$= G_m - \frac{G_m d_{m+1} \otimes \overline{G_m d_{m+1}}}{\langle G_m d_{m+1}, d_{m+1}\rangle}, \tag{5.190}$$

which implies Eq.(5.90). ■

5.8.12 Theorem 5.25

For any ONB $\{\varphi_p\}_{p=1}^\mu$ in H, it holds that

$$\sum_{p=1}^{\mu} (\varphi_p \otimes \overline{\varphi_p}) = I_H, \tag{5.191}$$

where I_H is the identity operator on H. Hence, if $\{\sqrt{k}\phi_m\}_{m=1}^M$ consists of k sets of ONBs, it follows from Eq.(5.121) that

$$\begin{aligned} Y^*Y &= \sum_{m=1}^{M} \left(\phi_m \otimes \overline{\phi_m}\right) = \frac{1}{k}\sum_{m=1}^{M} \left(\sqrt{k}\phi_m \otimes \overline{\sqrt{k}\phi_m}\right) \\ &= I_H. \end{aligned} \tag{5.192}$$

According to Items 1 and 2 in Proposition 5.22, Eq.(5.192) is equivalent to that a set $\{\phi_m\}_{m=1}^M$ forms a POB in H. In this case, the set $\{\phi_m\}_{m=1}^M$ is a PONB in H since $\|\phi_m\| = \frac{1}{\sqrt{k}}$ for $m = 1, 2, \ldots, M$. ■

Chapter 6

Theory of active learning with model selection

6.1 Introduction

In Chapters 4 and 5, the problems of active learning and model selection have been independently studied. If sample points and models are simultaneously optimized, then a higher level of the generalization capability is expected to be acquired. We call this problem *active learning with model selection.*

In general, the model should be fixed for active learning, and conversely the training examples $\{(\boldsymbol{x}_m, y_m)\}_{m=1}^M$ gathered at fixed sample points $\{\boldsymbol{x}_m\}_{m=1}^M$ are required for model selection. This implies that the problem of active learning with model selection can not be generally solved by simply combining existing active learning and model selection techniques. We call the fact the *active learning / model selection dilemma.*

In this chapter, we give a basic strategy for simultaneously optimizing sample points and models, and based on the strategy, we give a practical algorithm for active learning with model selection in trigonometric polynomial models.

6.2 Problem formulation

Let $\mathcal{X}$ be a set of M sample points $\{\boldsymbol{x}_m\}_{m=1}^M$, and let θ be a model. Let us denote the learning result function obtained with $\mathcal{X}$ and θ by $\hat{f}_{\mathcal{X},\theta}(\boldsymbol{x})$. We assume that the learning target function $f(\boldsymbol{x})$ and the learning result function $\hat{f}_{\mathcal{X},\theta}(\boldsymbol{x})$ belong to a specified reproducing kernel Hilbert space H (see Section 2.3), and the generalization error of

$\hat{f}_{\mathcal{X},\theta}(\boldsymbol{x})$ is measured by

$$J_G[\mathcal{X},\theta] = \mathrm{E}_{\boldsymbol{\epsilon}}\|\hat{f}_{\mathcal{X},\theta} - f\|^2, \tag{6.1}$$

where $\mathrm{E}_{\boldsymbol{\epsilon}}$ denotes the expectation over the noise. The norm is typically defined as

$$\|\hat{f}_{\mathcal{X},\theta} - f\|^2 = \int \left|\hat{f}_{\mathcal{X},\theta}(\boldsymbol{u}) - f(\boldsymbol{u})\right|^2 w(\boldsymbol{u})d\boldsymbol{u}, \tag{6.2}$$

where the integral with respect to $\boldsymbol{u}$ means the expectation over future sample points $\boldsymbol{u}$ and $w(\boldsymbol{u})$ is some weight function, e.g., the probability density function of $\boldsymbol{u}$. Then the problem of active learning with model selection considered in this chapter is formulated as follows.

Definition 6.1 (Active learning with model selection) *Determine a set $\mathcal{X}$ of sample points and select a model θ from a set $\mathcal{M}$ of model candidates so that the generalization error J_G is minimized:*

$$(\hat{\mathcal{X}},\hat{\theta}) = \operatorname*{argmin}_{\mathcal{X},\theta\in\mathcal{M}} J_G[\mathcal{X},\theta]. \tag{6.3}$$

6.3 Basic strategy

As we pointed out in Section 6.1, the problem of active learning with model selection can not be generally solved by simply combining existing active learning and model selection techniques because of the active learning / model selection dilemma: the model should be fixed for active learning and conversely sample points should be fixed for model selection.

However, if there is a set $\mathcal{X}$ of sample points that is optimal for all models in the set $\mathcal{M}$, the problem of active learning with model selection can be straightforwardly solved as follows. First, $\mathcal{X}$ is determined so that it is optimal for all models in the set $\mathcal{M}$, and sample values $\{y_m\}_{m=1}^M$ are gathered at the optimal points $\{\boldsymbol{x}_m\}_{m=1}^M$. Then model selection is performed with the optimal training examples $\{(\boldsymbol{x}_m, y_m)\}_{m=1}^M$. Consequently, we obtain the optimal model with optimal sample points because the sample points are optimal for any selected model. This basic strategy is summarized in Figure 6.1.

In the following section, we give a procedure for active learning with model selection based on the above strategy.

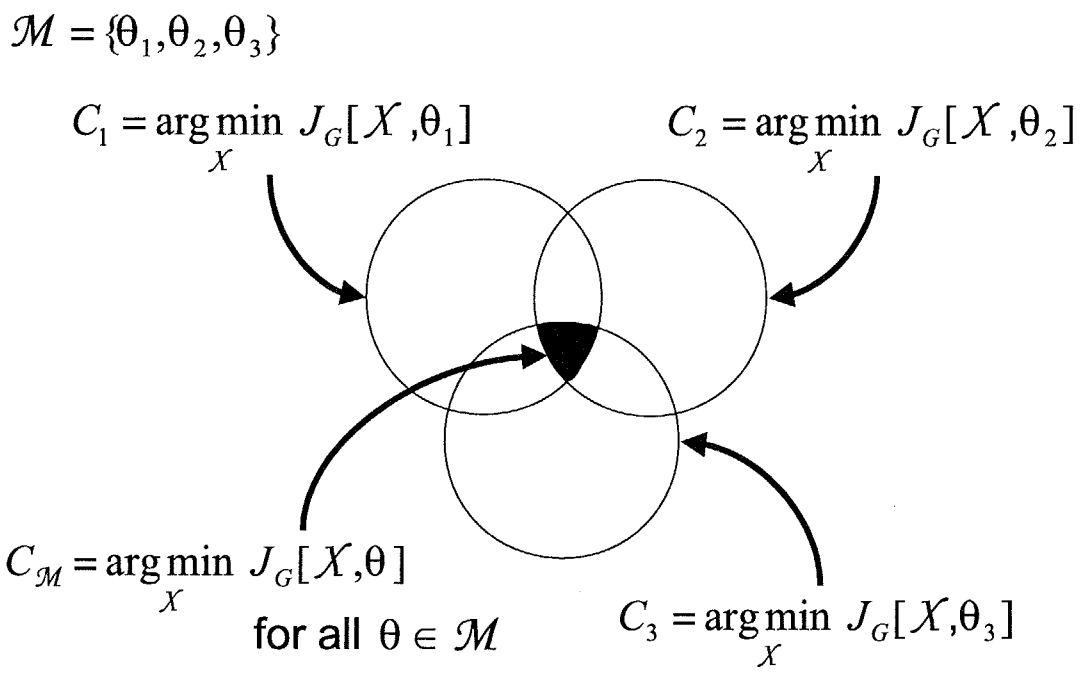


Figure 6.1: Basic strategy for active learning with model selection. Let the set $\mathcal{M}$ of models be $\{\theta_1, \theta_2, \theta_3\}$. The top-left circle denotes a set C_1 of optimal $\mathcal{X}$ for the model θ_1, i.e., an element in C_1 is a set $\mathcal{X}$ of sample points $\{\boldsymbol{x}_m\}_{m=1}^M$ that minimizes $J_G[\mathcal{X}, \theta_1]$. Similarly, the top-right and bottom circles denote sets of optimal $\mathcal{X}$ for θ_2 and θ_3, respectively. If there exists $\mathcal{X}$ that is commonly optimal for all models in $\mathcal{M}$, i.e., $C_{\mathcal{M}}$ is not empty, then the problem of active learning with model selection can be straightforwardly solved by using the commonly optimal sample points.

6.4 Active learning with model selection for trigonometric polynomial models

In this section, we give a procedure for simultaneously optimizing sample points and models. For simplicity, we focus on the case when the dimension L of the input vector $\boldsymbol{x}$ is one. However, all the discussions in this section can be scaled to the case when $L > 1$.

6.4.1 Setting

First, the setting is described.

1. The function space H to which the learning target function $f(x)$ belongs is S_N, a trigonometric polynomial space of order N (see Section 3.3.1):

$$H = S_N. \tag{6.4}$$

In this case, the generalization error J_G is expressed as

$$J_G[\mathcal{X}, \theta] = \mathrm{E}_{\boldsymbol{\epsilon}} \frac{1}{2\pi} \int \left| \hat{f}_{\mathcal{X},\theta}(u) - f(u) \right|^2 du, \tag{6.5}$$

i.e., the weight function $w(u)$ in Eq.(6.2) is assigned to $1/2\pi$.

2. The number M of training examples is larger than the dimension of S_N:

$$M > \dim S_N = 2N + 1. \tag{6.6}$$

3. LMS learning is adopted (see Section 3.2.1). In this case, a model θ indicates a subspace S. The learning result function $\hat{f}_{\mathcal{X},S_n}(x)$ obtained with the sample points $\mathcal{X}$ and model S_n is given as

$$\hat{f}_{\mathcal{X},S_n} = A^{\dagger}_{\mathcal{X},S_n} \boldsymbol{y}, \tag{6.7}$$

where $A_{\mathcal{X},S_n}$ is an operator from H to $\mathbf{C}^M$ defined with the reproducing kernel $K_{S_n}(x, x')$ of S_n as

$$A_{\mathcal{X},S_n} = \sum_{m=1}^{M} \left(e_m \otimes \overline{K_{S_n}(\cdot, x_m)} \right). \tag{6.8}$$

Here, $(\cdot \otimes \bar{\cdot})$ denotes the Neumann-Schatten product (see Section 2.2.4) and $\boldsymbol{e}_m$ is the m-th vector of the so-called standard basis in $\mathbf{C}^M$. The vector $\boldsymbol{y}$ is defined as

$$\boldsymbol{y} = (y_1, y_2, \ldots, y_M)^{\top}, \tag{6.9}$$

where $\top$ denotes the transpose of a vector. It holds for any function f in S_N that

$$A_{\mathcal{X},S_N} f = \boldsymbol{z}, \tag{6.10}$$

where $\boldsymbol{z}$ is defined as

$$\boldsymbol{z} = (f(x_1), f(x_2), \ldots, f(x_M))^{\top}. \tag{6.11}$$

This can be verified from the property of the reproducing kernel (see Section 2.3):

$$\langle f(\cdot), K_{S_N}(\cdot, x') \rangle = f(x'). \tag{6.12}$$

4. The set $\mathcal{M}$ of model candidates consists of all trigonometric polynomial spaces included in S_N:

$$\mathcal{M} = \{ S_n \mid n = 0, 1, \ldots, N \}. \tag{6.13}$$

5. For any sample points $\{x_m\}_{m=1}^M$, the mean noise is zero:

$$\mathrm{E}_{\boldsymbol{\epsilon}}\boldsymbol{\epsilon} = 0, \tag{6.14}$$

where

$$\boldsymbol{\epsilon} = (\epsilon_1, \epsilon_2, \ldots, \epsilon_M)^\top. \tag{6.15}$$

6. For any sample points $\{x_m\}_{m=1}^M$, the noise covariance matrix Q is in the form

$$Q = \mathrm{E}_{\boldsymbol{\epsilon}}\left(\boldsymbol{\epsilon} \otimes \overline{\boldsymbol{\epsilon}}\right) = \sigma^2 I_M \tag{6.16}$$

with $\sigma^2 > 0$, where I_M is the M-dimensional identity matrix. σ^2 does not have to be known.

6.4.2 Procedure for active learning with model selection

Under the above setting, we will give a procedure for simultaneously optimizing sample points and models.

It is known that the generalization error of $\hat{f}_{\mathcal{X},S_n}(x)$ can be decomposed into the bias and variance (see e.g. Takemura [132], Geman *et al.* [40], Efron & Tibshirani [33]):

$$J_G[\mathcal{X}, S_n] = \|\mathrm{E}_{\boldsymbol{\epsilon}}\hat{f}_{\mathcal{X},S_n} - f\|^2 + \mathrm{E}_{\boldsymbol{\epsilon}}\|\hat{f}_{\mathcal{X},S_n} - \mathrm{E}_{\boldsymbol{\epsilon}}\hat{f}_{\mathcal{X},S_n}\|^2. \tag{6.17}$$

Note that the bias of $\hat{f}_{\mathcal{X},S_n}(x)$ can not be zero unless the learning target function $f(x)$ belongs to S_n.

As shown in Theorem 5.2 in page 95, a set $\mathcal{X}$ of sample points $\{x_m\}_{m=1}^M$ minimizes the variance under the constraint of the bias being zero for a model S_n to which the learning target function $f(x)$ belongs if and only if $\frac{1}{M}A^*_{\mathcal{X},S_n}A_{\mathcal{X},S_n}$ agrees with the identity operator on S_n. Since we consider the function space H $(= S_N \supset S_n)$ in this section, the above optimality condition is expressed as

$$\frac{1}{M}A^*_{\mathcal{X},S_n}A_{\mathcal{X},S_n} = P_{S_n}, \tag{6.18}$$

where P_{S_n} denotes the orthogonal projection operator onto S_n in H.

It is shown in Theorems 5.8 and 5.9 in pages 101 and 101, respectively, that there are infinitely many sets of sample points that satisfy Condition (6.18) for a fixed model S_n. Here, we show a design method of sample points that satisfy Condition (6.18) for all models in the set $\mathcal{M}$.

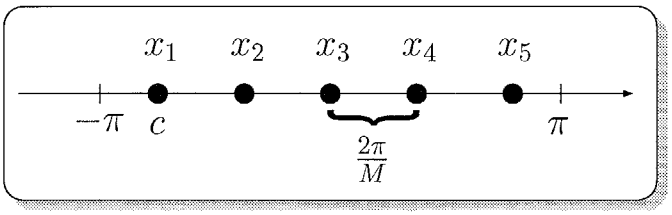


Figure 6.2: Commonly optimal sample points for all trigonometric polynomial models designed by Eq.(6.20). The largest order N of the trigonometric polynomial space is 1 and the number M of training examples is 5.

Theorem 6.2 *Let $M \geq 2N+1$ and c be an arbitrary constant such that*

$$-\pi \leq c \leq -\pi + \frac{2\pi}{M}. \tag{6.19}$$

If a set $\{x_m\}_{m=1}^M$ of sample points is fixed to

$$x_m = c + \frac{2\pi}{M}(m-1), \tag{6.20}$$

then it holds that

$$\frac{1}{M} A^*_{\mathcal{X},S_n} A_{\mathcal{X},S_n} = P_{S_n} \quad \text{for all } S_n \in \mathcal{M}. \tag{6.21}$$

Theorem 6.2 is clear from Theorems 5.2 and 5.8, so we omit the proof.

Eq.(6.20) means that M sample points are fixed to regular intervals in the domain (see Figure 6.2). Theorems 5.2 and 6.2 asserts that the sample points designed by Eq.(6.20) are optimal for all models to which the learning target function $f(x)$ belongs.

With training examples $\{(x_m, y_m)\}_{m=1}^M$ gathered at the optimal sample points $\{x_m\}_{m=1}^M$ designed by Eq.(6.20), we will perform model selection. As a model selection criterion, we use SIC proposed in Chapter 4. SIC for the present setting is given as follows.

$$\begin{aligned} \mathrm{SIC}[S_n] = \|(A^\dagger_{S_n} - A^\dagger_{S_N})\boldsymbol{y}\|^2 - \hat{\sigma}^2 \mathrm{tr}(A^\dagger_{S_n} - A^\dagger_{S_N})(A^\dagger_{S_n} - A^\dagger_{S_N})^* \\ + \hat{\sigma}^2 \mathrm{tr} A^\dagger_{S_n}(A^\dagger_{S_n})^*, \end{aligned} \tag{6.22}$$

where

$$A_{S_n} = \sum_{m=1}^M \left(\boldsymbol{e}_m \otimes \overline{K_{S_n}(\cdot, x_m)}\right), \tag{6.23}$$

$$A_{S_N} = \sum_{m=1}^M \left(\boldsymbol{e}_m \otimes \overline{K_{S_N}(\cdot, x_m)}\right), \tag{6.24}$$

$$\hat{\sigma}^2 = \frac{\sum_{m=1}^{M} \left| \hat{f}_{S_N}(x_m) - y_m \right|^2}{M - \dim S_N}. \tag{6.25}$$

6.4.3 Exact and fast algorithm for active learning with model selection

Now we give an exact and fast algorithm of active learning with model selection for trigonometric polynomial models.

For general sample points, SIC is calculated by Corollary 4.4 in page 37. When the sample points are designed by Eq.(6.20), the following lemma holds.

Lemma 6.3 *When Eq.(6.21) holds, SIC for a model S_n is expressed as*

$$\mathrm{SIC}[S_n] = J_{TE}^{S_n} + \frac{2\hat{\sigma}^2}{M} \dim S_n - \hat{\sigma}^2, \tag{6.26}$$

where $J_{TE}^{S_n}$ is the training error of $\hat{f}_{S_n}(x)$ defined as

$$J_{TE}^{S_n} = \frac{1}{M} \sum_{m=1}^{M} \left| \hat{f}_{S_n}(x_m) - y_m \right|^2. \tag{6.27}$$

$\hat{\sigma}^2$ is given as

$$\hat{\sigma}^2 = \frac{M}{M - \dim S_N} J_{TE}^{S_N}. \tag{6.28}$$

A proof of Lemma 6.3 is given in Section 6.6.1.

Note that Eq.(6.26) is equivalent to C_P (Mallows [72][73]). In Eq.(6.26), terms that depend on the model S_n are only $J_{TE}^{S_n}$ and $\dim S_n$. As regards $J_{TE}^{S_n}$, the following lemma holds.

Lemma 6.4 *When Eq.(6.21) holds, we have*

$$J_{TE}^{S_0} = \frac{1}{M} \sum_{m=1}^{M} |y_m|^2 - \left| \frac{1}{M} \sum_{m=1}^{M} y_m \right|^2, \tag{6.29}$$

$$J_{TE}^{S_n} = J_{TE}^{S_{n-1}} - \left| \frac{1}{M} \sum_{m=1}^{M} y_m \exp(inx_m) \right|^2 - \left| \frac{1}{M} \sum_{m=1}^{M} y_m \exp(-inx_m) \right|^2. \tag{6.30}$$

A proof of Lemma 6.4 is given in Section 6.6.2.

From Lemmas 6.3 and 6.4, we have the following theorem.

input M, N, and c such that
$M > 2N+1$ and $-\pi \leq c \leq -\pi + \frac{2\pi}{M}$;
for $m = 1, 2, \ldots, M$ {
$x_m \longleftarrow c + \frac{2\pi(m-1)}{M}$;
gather sample value y_m at x_m;
}
for $n = -N, -N+1, \ldots, N$ {
$a_n \longleftarrow |\frac{1}{M}\sum_{m=1}^{M} y_m \exp(inx_m)|^2$;
}
$\hat{\sigma}^2 \longleftarrow \frac{M}{M-2N-1}(\frac{1}{M}\sum_{m=1}^{M}|y_m|^2 - \sum_{n=-N}^{N} a_n)$;
$\mathrm{SIC}_0 \longleftarrow \frac{1}{M}\sum_{m=1}^{M}|y_m|^2 - a_0 + \frac{2\hat{\sigma}^2}{M} - \hat{\sigma}^2$;
for $n = 1, 2, \ldots, N$ {
$\mathrm{SIC}_n \longleftarrow \mathrm{SIC}_{n-1} - a_n - a_{-n} + \frac{4\hat{\sigma}^2}{M}$;
}
$\hat{n} \longleftarrow \mathrm{argmin}_n \mathrm{SIC}_n$;
$\hat{f}(x) \longleftarrow \sum_{p=-\hat{n}}^{\hat{n}} |\frac{1}{M}\sum_{m=1}^{M} y_m \exp(-ipx_m)|^2 \exp(ipx)$;

Figure 6.3: Exact and fast algorithm for active learning with model selection.

Theorem 6.5 *When Eq.(6.21) holds, we have*

$$\mathrm{SIC}[S_0] = J_{TE}^{S_0} + \frac{2\hat{\sigma}^2}{M} - \hat{\sigma}^2, \tag{6.31}$$

$$\mathrm{SIC}[S_n] = \mathrm{SIC}[S_{n-1}] - \left|\frac{1}{M}\sum_{m=1}^{M} y_m \exp(inx_m)\right|^2 - \left|\frac{1}{M}\sum_{m=1}^{M} y_m \exp(-inx_m)\right|^2 + \frac{4\hat{\sigma}^2}{M}. \tag{6.32}$$

A proof of Theorem 6.5 is given in Section 6.6.3.

Based on Theorem 6.5, an exact and fast algorithm for active learning with model selection is described in Figure 6.3.

Let us measure the computational complexity by the number of scalar multiplications (Table 6.1). When SIC is calculated by Corollary 4.4, the computational complexity and memory required for active learning with model selection are $O(N^3(M+N))$ and $O(M+N^2)$, respectively. In contrast, they are reduced to $O(MN)$ and $O(M+N)$ when the algorithm shown in Figure 6.3 is used. This shows that the proposed algorithm is exact and much more efficient than the straightforward calculation.

Table 6.1: Computational complexity and memory required for active learning with model selection.

	Computational complexity	Memory
Straightforward calculation (Corollary 4.4)	$O(N^3(M+N))$	$O(M+N^2)$
Exact and fast algorithm (Figure 6.3)	$O(MN)$	$O(M+N)$

6.5 Computer simulations

In this section, the effectiveness of the proposed procedure for active learning with model selection is demonstrated through computer simulations.

6.5.1 Setting

Let us consider the chaotic series created by the *Mackey-Glass delay-difference equation* (see e.g. Platt [97]):

$$g(t+1) = \begin{cases} (1-b)g(t) + \dfrac{a\, g(t-\tau)}{1+g(t-\tau)^{10}} & \text{for } t \geq \tau+1, \\ 0.3 & \text{for } 0 \leq t \leq \tau, \end{cases} \tag{6.33}$$

where $a = 0.2$, $b = 0.1$, and $\tau = 17$. Let $\{h_t\}_{t=1}^{600}$ be

$$h_t = g(t+\tau+1). \tag{6.34}$$

We are given M degraded sample values $\{y_m\}_{m=1}^{M}$:

$$y_m = h_{r(m)} + \epsilon_m, \tag{6.35}$$

where $r(m)$ is an integer such that $1 \leq r(m) \leq 600$ which indicates the sampling location, and the noise ϵ_m is independently subject to the same normal distribution with mean 0 and variance σ^2:

$$\epsilon_m \sim N(0, \sigma^2). \tag{6.36}$$

The task is to obtain the best estimates $\{\hat{h}_t\}_{t=1}^{600}$ of $\{h_t\}_{t=1}^{600}$ that minimizes the error:

$$\text{Error} = \frac{1}{600}\sum_{t=1}^{600}\left|\hat{h}_t - h_t\right|^2. \tag{6.37}$$

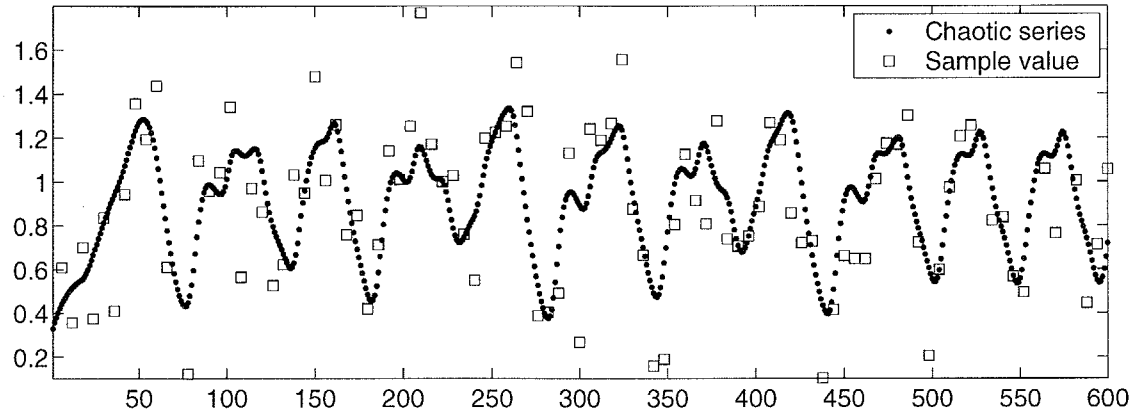


Figure 6.4: Chaotic series and 100 sample values ($r(m) = 6m$ and $\sigma^2 = 0.07$).

In this simulation, we consider four cases when $(M, \sigma^2) = (300, 0.04)$, $(100, 0.04)$, $(300, 0.07)$, and $(100, 0.07)$. Figure 6.4 displays the original chaotic series $\{h_t\}_{t=1}^{600}$ (shown by '•') and an example of 100 sample values $\{y_m\}_{m=1}^{100}$ (shown by '□') with the noise variance $\sigma^2 = 0.07$.

We shall obtain the estimates $\{\hat{h}_t\}_{t=1}^{600}$ as follows. Let us consider sample points $\{x_m\}_{m=1}^{M}$ corresponding to the sample values $\{y_m\}_{m=1}^{M}$:

$$x_m = -\pi + \frac{2\pi}{600}(r(m) - 1). \tag{6.38}$$

By using the training examples $\{(x_m, y_m)\}_{m=1}^{M}$, we perform LMS learning. Then the LMS learning function $\hat{f}(x)$ gives the estimates $\{\hat{h}_t\}_{t=1}^{600}$ as

$$\hat{h}_t = \hat{f}\left(-\pi + \frac{2\pi}{600}(t-1)\right). \tag{6.39}$$

We adopt S_{40}, a trigonometric polynomial space of order 40 (see Section 3.3.1), as H. Note that the 600 chaotic series can not be expressed by the functions in S_{40}. This means that we consider the learning target function which is not included in H. Let the set $\mathcal{M}$ of model candidates be

$$\mathcal{M} = \{S_0, S_1, S_2, \ldots, S_{40}\}. \tag{6.40}$$

6.5.2 Active learning

First, we shall compare the performance of the following two sampling schemes.

(i) Optimal sampling: Sample points are fixed to regular intervals, i.e.,

$$r(m) = \frac{600m}{M}. \tag{6.41}$$

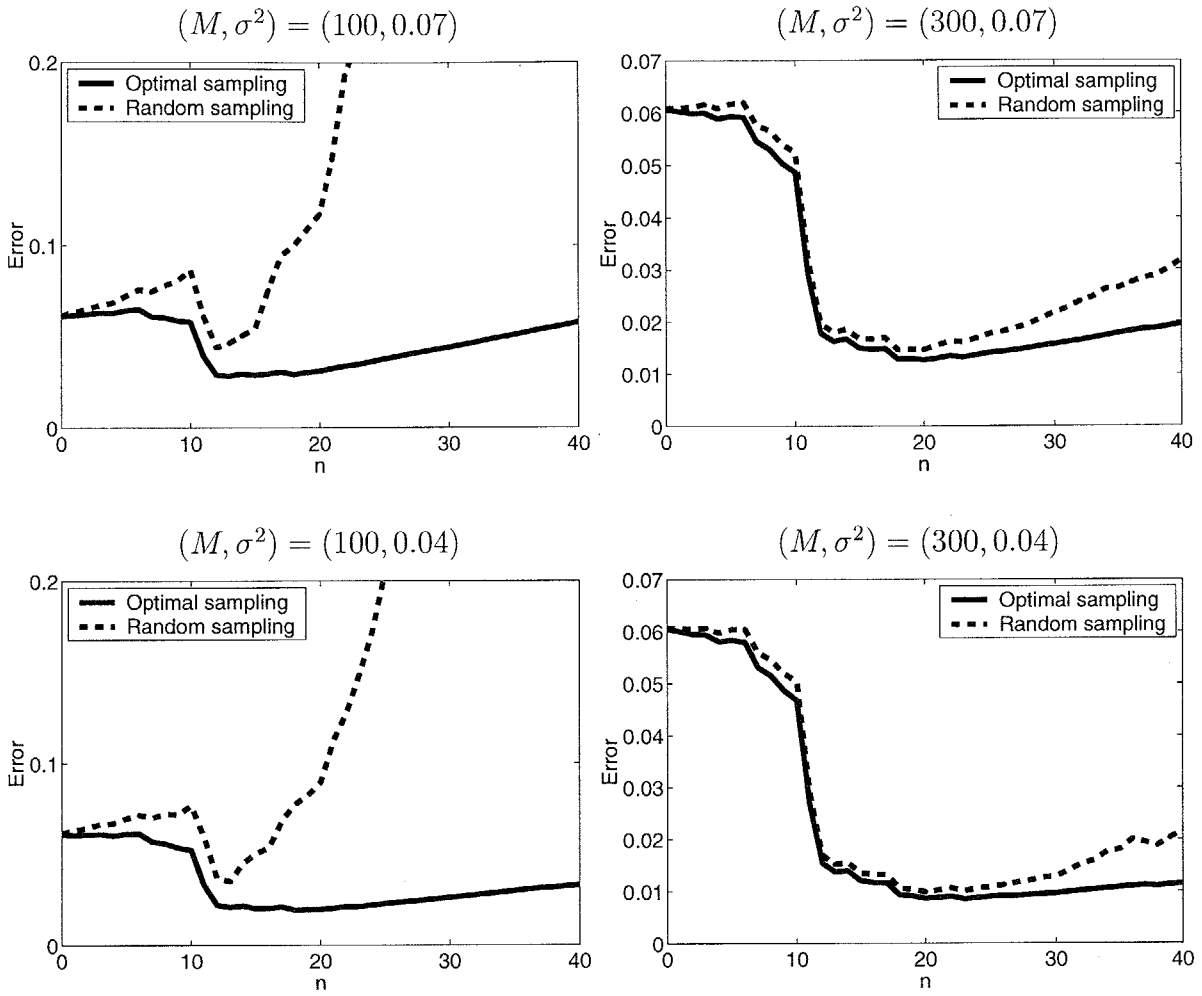


Figure 6.5: Results of active learning simulations.

In this case, Eqs.(6.38) and (6.41) yield Eq.(6.20) with $c = -\pi + \frac{2\pi}{M} - \frac{2\pi}{600}$.

(ii) Random sampling: Sample points are randomly created in the domain, i.e., $r(m)$ randomly gives an integer such that $1 \leq r(m) \leq 600$.

Figure 6.5 displays the results of the active learning simulation. The horizontal axis denotes the order n of the model and the vertical axis denotes the error measured by Eq.(6.37). The solid and dashed lines show the mean errors of 100 trials by (i) Optimal sampling and (ii) Random sampling, respectively. These graphs show that (i) Optimal sampling provides better generalization capability than (ii) Random sampling irrespective of the number M of training examples, noise variance σ^2, and order n of the model. Especially, when M is small and σ^2 is large (the top-left graph in Figure 6.5), its effectiveness is remarkable.

6.5.3 Model selection

By using the optimal sample points $\{x_m\}_{m=1}^M$ designed by Eqs.(6.38) and (6.41), we shall compare the performance of the following model selection criteria.

(a) Subspace information criterion (SIC): SIC is calculated by the algorithm shown in Figure 6.3.

(b) Leave-one-out cross-validation (CV): A closed form expression of the leave-one-out error for S_n is given as (see Orr [96])

$$\mathrm{CV}[S_n] = \frac{1}{M}\|(\mathrm{diag}(I_M - B_{S_n}B_{S_n}^\dagger))^{-1}(I_M - B_{S_n}B_{S_n}^\dagger)\boldsymbol{y}\|^2, \tag{6.42}$$

where the matrix 'diag$(I_M - B_{S_n}B_{S_n}^\dagger)$' is the same size and has the same diagonal as $(I_M - B_{S_n}B_{S_n}^\dagger)$ but is zero off the diagonal. B_{S_n} is the $M \times \mu$ matrix with the (m,p)-th element being

$$[B_{S_n}]_{m,p} = \begin{cases} \exp(\frac{ipx_m}{2}) & \text{if } p \leq 2n \text{ and } p \text{ is even,} \\ \exp(-\frac{i(p-1)x_m}{2}) & \text{if } p \leq 2n+1 \text{ and } p \text{ is odd,} \\ 0 & \text{if } p \geq 2n+2. \end{cases} \tag{6.43}$$

Note that $B_{S_n}^\dagger = \frac{1}{M}B_{S_n}^*$ when the sample points $\{x_m\}_{m=1}^M$ are designed by Eqs.(6.38) and (6.41).

(c) Akaike's information criterion (AIC) (Akaike [1]): When the noise is subject to the normal distribution, AIC for S_n is expressed as

$$\mathrm{AIC}[S_n] = M\log J_{TE}^{S_n} + 2(2n+1+1), \tag{6.44}$$

where $J_{TE}^{S_n}$ is the training error defined by Eq.(6.27).

(d) Corrected AIC (cAIC) (Sugiura [127]): When the noise is subject to the normal distribution, cAIC for S_n is expressed as

$$\mathrm{cAIC}[S_n] = M\log J_{TE}^{S_n} + \frac{2(2n+1+1)M}{M-(2n+1)-2}. \tag{6.45}$$

(e) Bayesian information criterion (BIC) (Schwarz [115]): When the noise is subject to the normal distribution, BIC for S_n is expressed as

$$\mathrm{BIC}[S_n] = M\log J_{TE}^{S_n} + (2n+1+1)\log M. \tag{6.46}$$

Note that the minimum description length (MDL) criterion (Rissanen [99][100][101]) is the same expression as BIC.

(f) Vapnik's measure (VM) (Cherkassky *et al.* [23]): VM for S_n is given as

$$\mathrm{VM}[S_n] = J_{TE}^{S_n} \Big/ \max\left(0, 1 - \sqrt{p - p\log p + \frac{\log M}{2M}}\right), \tag{6.47}$$

where

$$p = \frac{2n+1}{M}. \tag{6.48}$$

Note that AIC, cAIC, BIC, MDL, and VM can be exactly and efficiently calculated by using Lemma 6.4.

Figures 6.6, 6.7, 6.8, and 6.9 display the simulation results. The top seven graphs show the values of the error and model selection criteria corresponding to the order n of the model S_n (see Eq.(6.40)). The box plot notation specifies marks at 95, 75, 50, 25, and 5 percentiles of values. The solid line denotes the mean values. The bottom-left seven graphs show the distributions of the selected order n of models. 'OPT' indicates the optimal model that minimizes the error defined by Eq.(6.37). The bottom-right seven graphs show the distributions of the error obtained by the model selected by each criterion.

When $(M, \sigma^2) = (300, 0.04)$ (Figure 6.6), all model selection criteria work well. Figure 6.10 displays the target chaotic series $\{h_t\}_{t=1}^{600}$, sample values $\{y_m\}_{m=1}^{M}$, and SIC estimates $\{\hat{h}_t\}_{t=1}^{600}$. In this case, SIC selects S_{23} and the error measured by Eq.(6.37) is 6.80×10^{-3}.

When $(M, \sigma^2) = (100, 0.04)$ (Figure 6.7), SIC, CV, and cAIC work well. In contrast, AIC tends to select larger models, and BIC (MDL) and VM are inclined to select smaller models. Consequently, they yield large errors.

When $(M, \sigma^2) = (300, 0.07)$ (Figure 6.8), SIC, CV, AIC, and cAIC work well. Although BIC (MDL) and VM also work well on the whole, they sometimes select the smallest model and provide large errors.

Finally, when $(M, \sigma^2) = (100, 0.07)$ (Figure 6.9), SIC and CV almost always selects reasonable models, so they provide small errors. In contrast, AIC tends to select larger models, and cAIC, BIC, and VM tend to select smaller models. As a result, they give large errors. Figure 6.11 displays the target chaotic series $\{h_t\}_{t=1}^{600}$, sample values $\{y_m\}_{m=1}^{M}$, and SIC estimates $\{\hat{h}_t\}_{t=1}^{600}$. In this case, SIC selects S_{13} and the error measured by Eq.(6.37) is 2.33×10^{-2}.

This simulation shows that SIC gives a very good estimate of the error on average even when the learning target function is not included in H. As a result, SIC works well even with a small number M of training examples and a large noise variance σ^2.

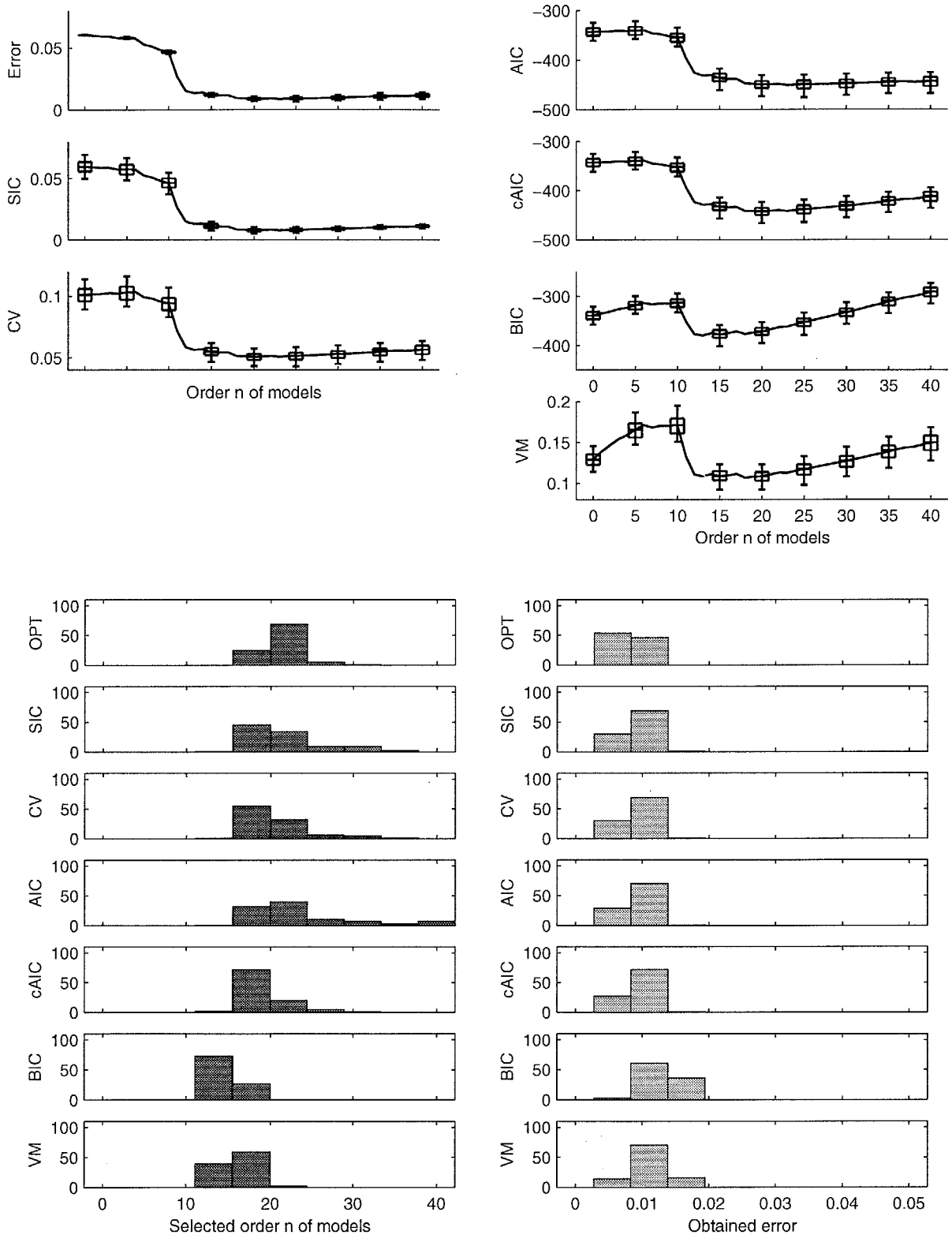


Figure 6.6: Results of model selection simulation when $(M, \sigma^2) = (300, 0.04)$.

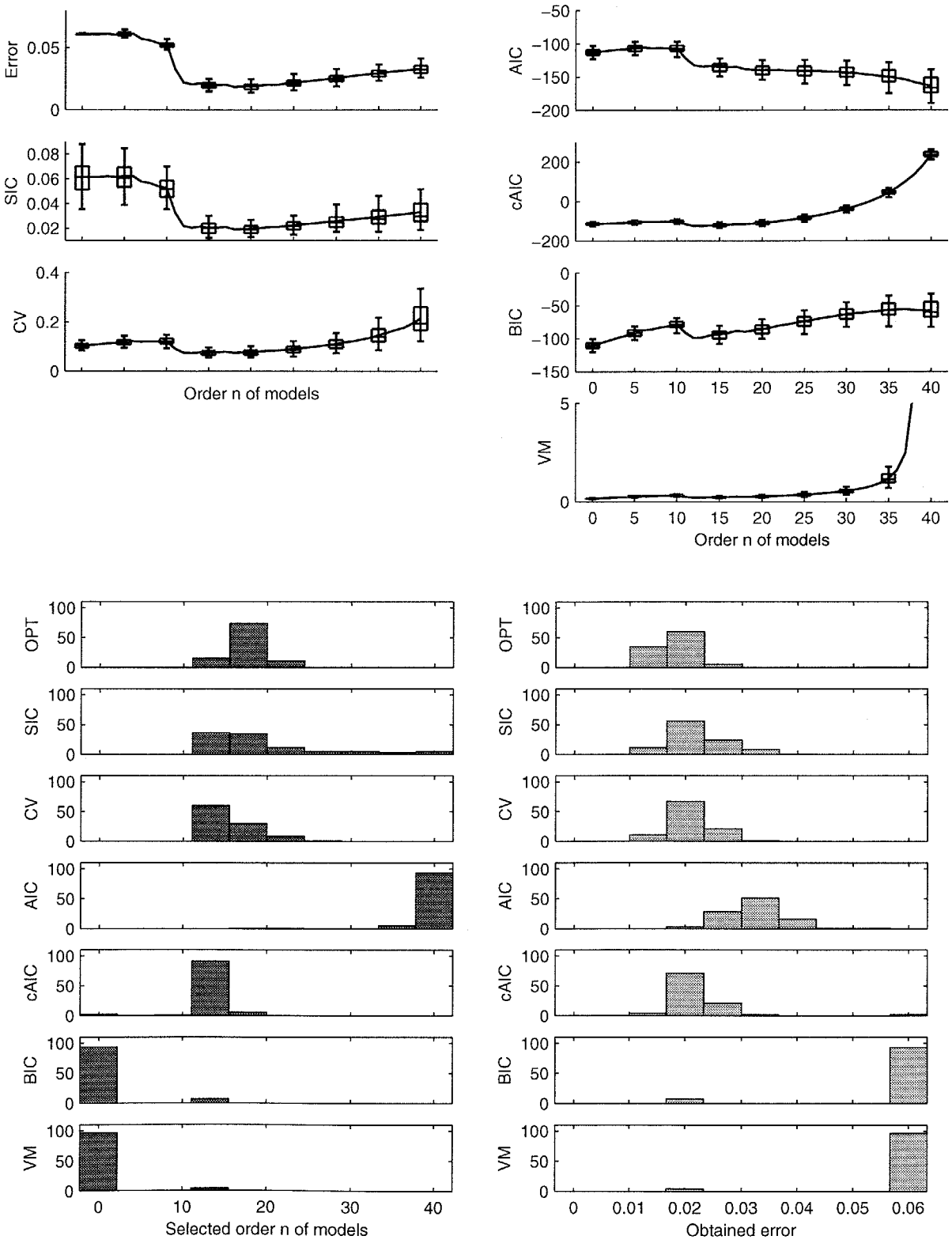


Figure 6.7: Results of model selection simulation when $(M, \sigma^2) = (100, 0.04)$.

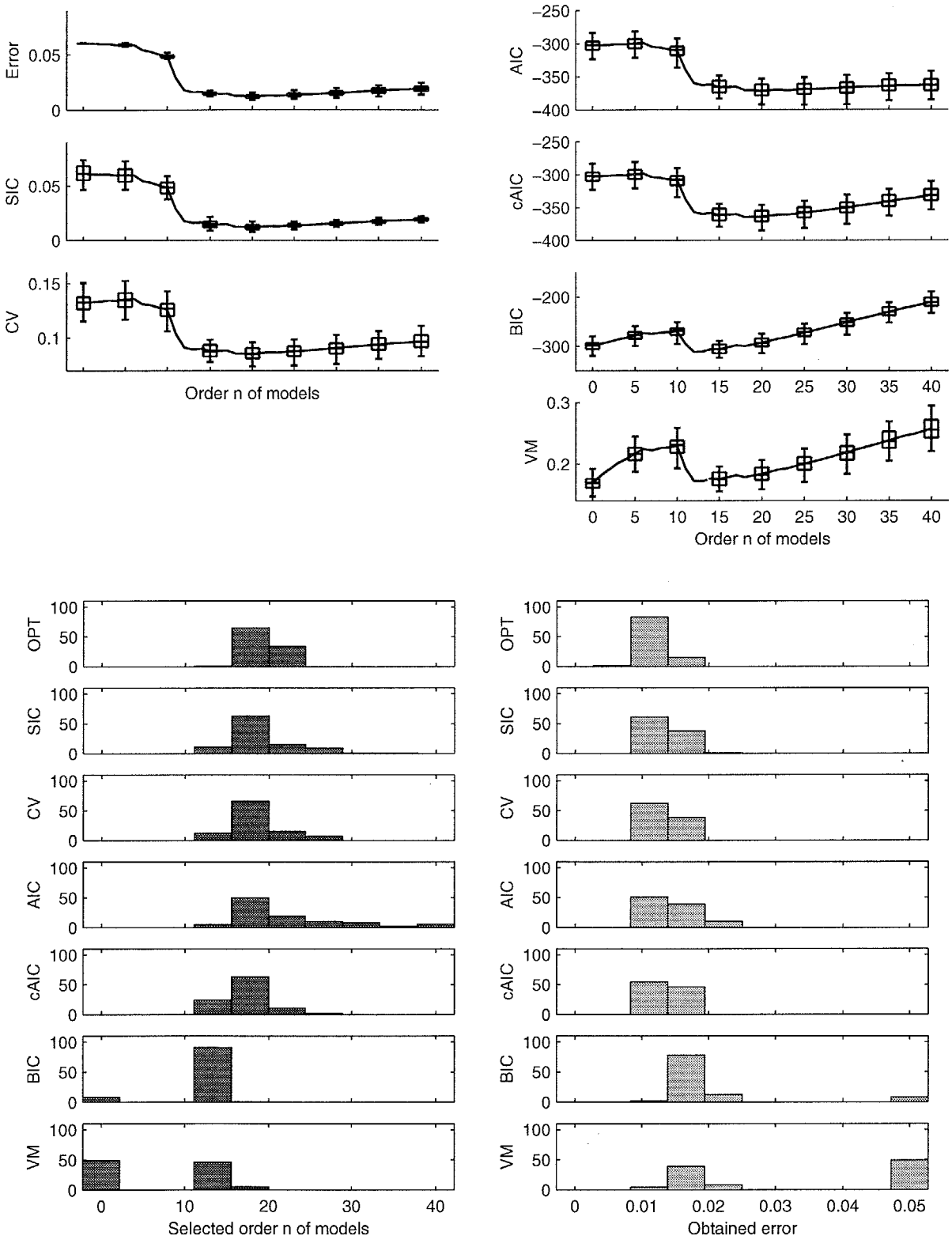


Figure 6.8: Results of model selection simulation when $(M, \sigma^2) = (300, 0.07)$.

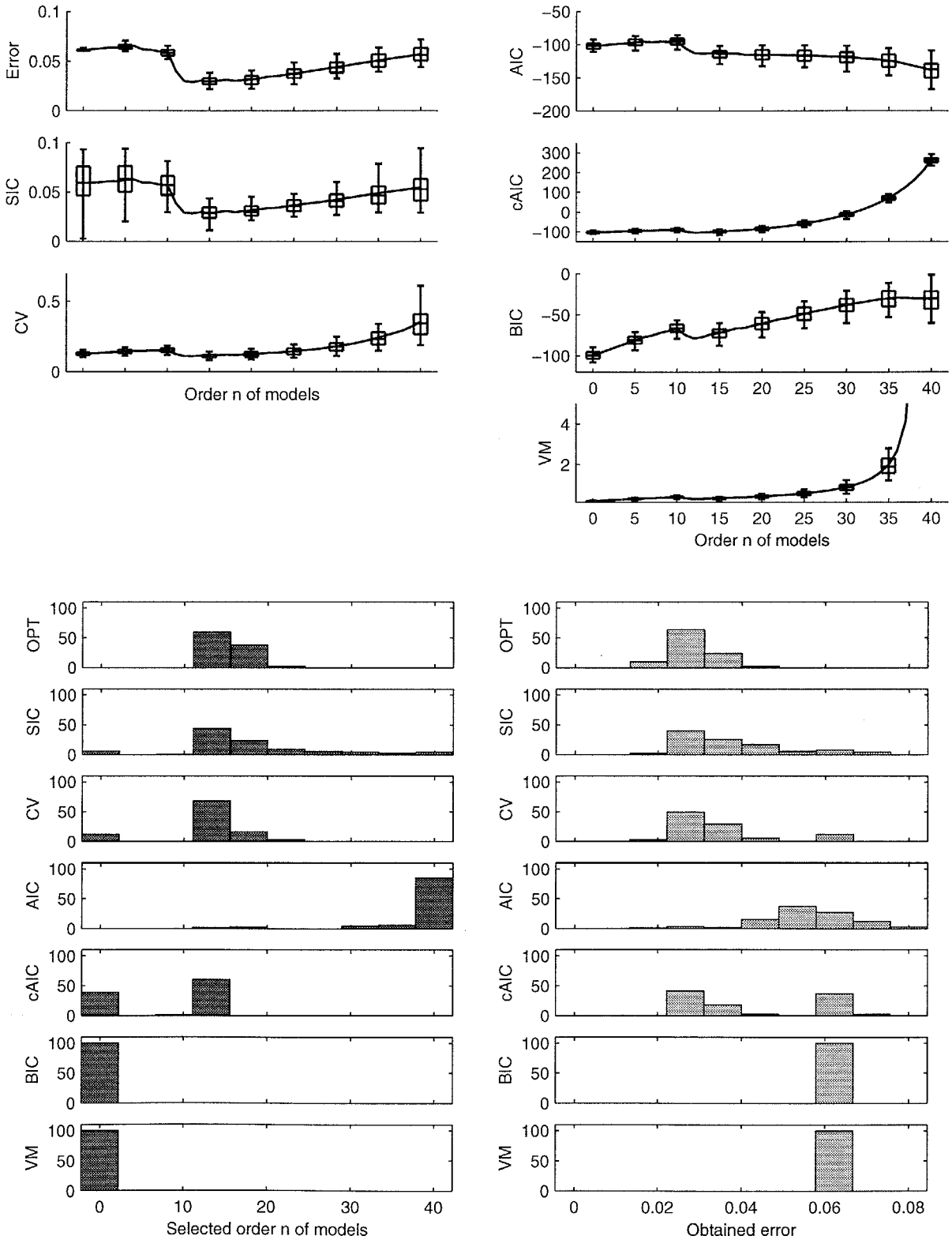


Figure 6.9: Results of model selection simulation when $(M, \sigma^2) = (100, 0.07)$.

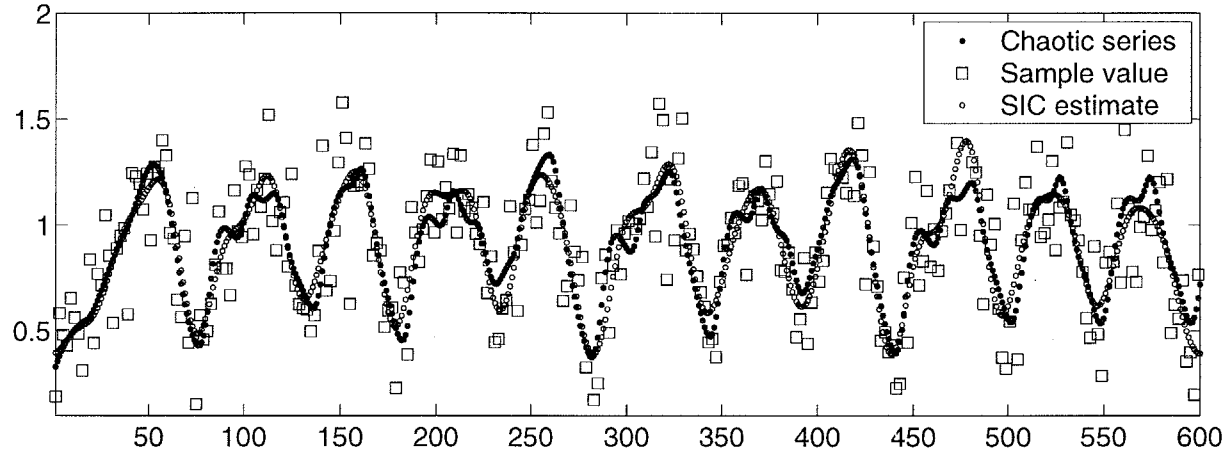


Figure 6.10: SIC estimates of chaotic series when $(M, \sigma^2) = (300, 0.04)$. S_{23} is selected and the error measured by Eq.(6.37) is 6.80×10^{-3}.

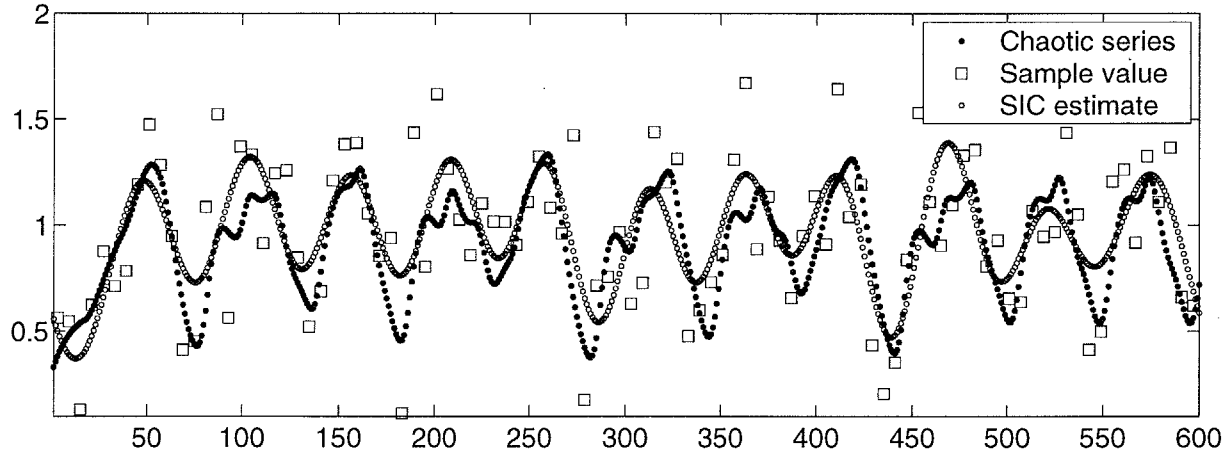


Figure 6.11: SIC estimates of chaotic series when $(M, \sigma^2) = (100, 0.07)$. S_{13} is selected and the error measured by Eq.(6.37) is 2.33×10^{-2}.

6.6 Proofs

In this section, proofs of all theorems and lemmas given in this chapter are provided.

6.6.1 Lemma 6.3

It follows from Eq.(6.12) that

$$\begin{aligned} K_{S_n}(x_m, x_{m'}) &= \langle K_{S_n}(\cdot, x_{m'}), K_{S_n}(\cdot, x_m)\rangle \\ &= \langle K_{S_N}(\cdot, x_{m'}), K_{S_n}(\cdot, x_m)\rangle \\ &= \langle K_{S_n}(\cdot, x_{m'}), K_{S_N}(\cdot, x_m)\rangle. \end{aligned} \tag{6.49}$$

Therefore, it follows from Eqs.(6.23) and (6.24) that

$$A_{S_n}A_{S_n}^* = A_{S_n}A_{S_N}^* = A_{S_N}A_{S_n}^*. \tag{6.50}$$

When Eq.(6.21) holds, $A_{S_n}^\dagger$ is expressed as

$$\begin{aligned} A_{S_n}^\dagger &= (A_{S_n}^* A_{S_n})^\dagger A_{S_n}^* = (MP_{S_n})^\dagger A_{S_n}^* \\ &= \frac{1}{M} A_{S_n}^*. \end{aligned} \tag{6.51}$$

Then it follows from Eqs.(6.51) and (6.50) that

$$\begin{aligned} (A_{S_n}^\dagger - A_{S_N}^\dagger)^*(A_{S_n}^\dagger - A_{S_N}^\dagger) &= (A_{S_n}^\dagger)^* A_{S_n}^\dagger - (A_{S_n}^\dagger)^* A_{S_N}^\dagger - (A_{S_N}^\dagger)^* A_{S_n}^\dagger + (A_{S_N}^\dagger)^* A_{S_N}^\dagger \\ &= \frac{1}{M^2}(A_{S_n}A_{S_n}^* - A_{S_n}A_{S_N}^* - A_{S_N}A_{S_n}^* + A_{S_N}A_{S_N}^*) \\ &= \frac{1}{M^2}(A_{S_n}A_{S_n}^* - A_{S_n}A_{S_n}^* - A_{S_n}A_{S_n}^* + A_{S_N}A_{S_N}^*) \\ &= \frac{1}{M^2}(A_{S_N}A_{S_N}^* - A_{S_n}A_{S_n}^*). \end{aligned} \tag{6.52}$$

It follows from Eqs.(6.52) and (6.21) that

$$\begin{aligned} \mathrm{tr}(A_{S_n}^\dagger - A_{S_N}^\dagger)^*(A_{S_n}^\dagger - A_{S_N}^\dagger) &= \frac{1}{M^2}\left(\mathrm{tr}A_{S_N}A_{S_N}^* - \mathrm{tr}A_{S_n}A_{S_n}^*\right) \\ &= \frac{1}{M^2}\left(\mathrm{tr}A_{S_N}^*A_{S_N} - \mathrm{tr}A_{S_n}^*A_{S_n}\right) \\ &= \frac{1}{M^2}\left(\mathrm{tr}MP_{S_N} - \mathrm{tr}MP_{S_n}\right) \\ &= \frac{1}{M}\left(\dim S_N - \dim S_n\right). \end{aligned} \tag{6.53}$$

It follows from Eqs.(6.51) and (6.21) that

$$\begin{aligned}\mathrm{tr}A_{S_n}^{\dagger}(A_{S_n}^{\dagger})^* &= \frac{1}{M^2}\mathrm{tr}A_{S_n}^* A_{S_n} = \frac{1}{M^2}\mathrm{tr}MP_{S_n} \\ &= \frac{1}{M}\dim S_n. \end{aligned} \tag{6.54}$$

It follows from Eqs.(6.27), (6.10), (6.9), (6.7), and (6.51) that

$$\begin{aligned} J_{TE}^{S_n} &= \frac{1}{M}\sum_{m=1}^{M}\left|\hat{f}_{S_n}(x_m) - y_m\right|^2 = \frac{1}{M}\|A_{S_n}\hat{f}_{S_n} - \boldsymbol{y}\|^2 \\ &= \frac{1}{M}\|A_{S_n}A_{S_n}^{\dagger}\boldsymbol{y} - \boldsymbol{y}\|^2 = \frac{1}{M}\|P_{\mathcal{R}(A_{S_n})}\boldsymbol{y} - \boldsymbol{y}\|^2 \\ &= \frac{1}{M}\left(\|\boldsymbol{y}\|^2 - \|P_{\mathcal{R}(A_{S_n})}\boldsymbol{y}\|^2\right) = \frac{1}{M}\left(\|\boldsymbol{y}\|^2 - \langle P_{\mathcal{R}(A_{S_n})}\boldsymbol{y}, \boldsymbol{y}\rangle\right) \\ &= \frac{1}{M}\left(\|\boldsymbol{y}\|^2 - \langle A_{S_n}A_{S_n}^{\dagger}\boldsymbol{y}, \boldsymbol{y}\rangle\right) = \frac{1}{M}\left(\|\boldsymbol{y}\|^2 - \frac{1}{M}\langle A_{S_n}A_{S_n}^{*}\boldsymbol{y}, \boldsymbol{y}\rangle\right) \\ &= \frac{\|\boldsymbol{y}\|^2}{M} - \frac{\langle A_{S_n}A_{S_n}^{*}\boldsymbol{y}, \boldsymbol{y}\rangle}{M^2}. \end{aligned} \tag{6.55}$$

It follows from Eqs.(6.25) and (6.27) that

$$\hat{\sigma}^2 = \frac{M}{M - \dim S_N}J_{TE}^{S_N}, \tag{6.56}$$

which implies Eq.(6.28). It follows from Eqs.(6.22), (6.52), (6.53), (6.54), (6.55), and (6.56) that

$$\begin{aligned} \mathrm{SIC}[S_n] &= \langle (A_{S_n}^{\dagger} - A_{S_N}^{\dagger})^*(A_{S_n}^{\dagger} - A_{S_N}^{\dagger})\boldsymbol{y}, \boldsymbol{y}\rangle - \hat{\sigma}^2\mathrm{tr}(A_{S_n}^{\dagger} - A_{S_N}^{\dagger})^*(A_{S_n}^{\dagger} - A_{S_N}^{\dagger}) \\ &\quad + \hat{\sigma}^2\mathrm{tr}A_{S_n}^{\dagger}(A_{S_n}^{\dagger})^* \\ &= \frac{\langle (A_{S_N}A_{S_N}^{*} - A_{S_n}A_{S_n}^{*})\boldsymbol{y}, \boldsymbol{y}\rangle}{M^2} - \hat{\sigma}^2\frac{\dim S_N - \dim S_n}{M} + \hat{\sigma}^2\frac{\dim S_n}{M} \\ &= \frac{\langle A_{S_N}A_{S_N}^{*}\boldsymbol{y}, \boldsymbol{y}\rangle}{M^2} - \frac{\langle A_{S_n}A_{S_n}^{*}\boldsymbol{y}, \boldsymbol{y}\rangle}{M^2} - \frac{\hat{\sigma}^2\dim S_N}{M} + \frac{2\hat{\sigma}^2\dim S_n}{M} \\ &= -\frac{\|\boldsymbol{y}\|^2}{M} + \frac{\langle A_{S_N}A_{S_N}^{*}\boldsymbol{y}, \boldsymbol{y}\rangle}{M^2} + \frac{\|\boldsymbol{y}\|^2}{M} - \frac{\langle A_{S_n}A_{S_n}^{*}\boldsymbol{y}, \boldsymbol{y}\rangle}{M^2} \\ &\quad - \frac{\hat{\sigma}^2\dim S_N}{M} + \frac{2\hat{\sigma}^2\dim S_n}{M} \\ &= -J_{TE}^{S_N} + J_{TE}^{S_n} - \frac{\hat{\sigma}^2\dim S_N}{M} + \frac{2\hat{\sigma}^2\dim S_n}{M} \\ &= -\frac{M - \dim S_N}{M}\hat{\sigma}^2 + J_{TE}^{S_n} - \frac{\hat{\sigma}^2\dim S_N}{M} + \frac{2\hat{\sigma}^2\dim S_n}{M} \\ &= J_{TE}^{S_n} + \frac{2\hat{\sigma}^2\dim S_n}{M} - \hat{\sigma}^2, \end{aligned} \tag{6.57}$$

which implies Eq.(6.26). ■

6.6.2 Lemma 6.4

Let a function $\varphi_p(x)$ be defined as

$$\varphi_p(x) = \exp(ipx) \quad \text{for } p = -n, -n+1, \ldots, n. \tag{6.58}$$

Since $\{\varphi_p(x)\}_{p=-n}^{n}$ is an orthonormal basis in S_n (see Section 3.3.1), the reproducing kernel of S_n is expressed as (see Section 2.3)

$$\begin{aligned} K_{S_n}(x, x') &= \sum_{p=-n}^{n} \varphi_p(x)\overline{\varphi_p(x')} \\ &= \sum_{p=-n}^{n} \exp(ipx)\exp(-ipx'). \end{aligned} \tag{6.59}$$

Then it follows from Eqs.(6.55), (6.23), and (6.59) that

$$\begin{aligned} J_{TE}^{S_n} &= \frac{1}{M}\|\boldsymbol{y}\|^2 - \frac{1}{M^2}\langle A_{S_n}A_{S_n}^{*}\boldsymbol{y}, \boldsymbol{y}\rangle \\ &= \frac{1}{M}\sum_{m=1}^{M}|y_m|^2 - \frac{1}{M^2}\sum_{m=1}^{M}\sum_{m'=1}^{M} K_{S_n}(x_m, x_{m'})\overline{y_m}y_{m'} \\ &= \frac{1}{M}\sum_{m=1}^{M}|y_m|^2 - \frac{1}{M^2}\sum_{m=1}^{M}\sum_{m'=1}^{M}\sum_{p=-n}^{n} \exp(ipx_m)\exp(-ipx'_m)\overline{y_m}y_{m'} \\ &= \frac{1}{M}\sum_{m=1}^{M}|y_m|^2 - \sum_{p=-n}^{n}\left|\frac{1}{M}\sum_{m=1}^{M} y_m \exp(-ipx_m)\right|^2. \end{aligned} \tag{6.60}$$

When $n = 0$, we have

$$J_{TE}^{S_0} = \frac{1}{M}\sum_{m=1}^{M}|y_m|^2 - \left|\frac{1}{M}\sum_{m=1}^{M} y_m\right|^2, \tag{6.61}$$

which implies Eq.(6.29). When $n \geq 1$, we have

$$\begin{aligned} J_{TE}^{S_n} &= \frac{1}{M}\sum_{m=1}^{M}|y_m|^2 - \sum_{p=-(n-1)}^{n-1}\left|\frac{1}{M}\sum_{m=1}^{M} y_m \exp(-ipx_m)\right|^2 \\ &\quad - \left|\frac{1}{M}\sum_{m=1}^{M} y_m \exp(inx_m)\right|^2 - \left|\frac{1}{M}\sum_{m=1}^{M} y_m \exp(-inx_m)\right|^2 \\ &= J_{TE}^{S_{n-1}} - \left|\frac{1}{M}\sum_{m=1}^{M} y_m \exp(inx_m)\right|^2 - \left|\frac{1}{M}\sum_{m=1}^{M} y_m \exp(-inx_m)\right|^2, \end{aligned} \tag{6.62}$$

which implies Eq.(6.30). ■

6.6.3 Theorem 6.5

It follows from Eq.(6.26) with $n = 0$ that

$$\mathrm{SIC}[S_0] = J_{TE}^{S_0} + \frac{2\hat{\sigma}^2 \dim S_0}{M} - \hat{\sigma}^2, \tag{6.63}$$

which implies Eq.(6.31). It follows from Eqs.(6.26) and (6.30) that

$$\begin{aligned}
\mathrm{SIC}[S_n] &= J_{TE}^{S_n} + \frac{2\hat{\sigma}^2 \dim S_n}{M} - \hat{\sigma}^2 \\
&= J_{TE}^{S_{n-1}} - \left| \frac{1}{M} \sum_{m=1}^{M} y_m \exp(inx_m) \right|^2 - \left| \frac{1}{M} \sum_{m=1}^{M} y_m \exp(-inx_m) \right|^2 \\
&\quad + \frac{2\hat{\sigma}^2(\dim S_{n-1} + 2)}{M} - \hat{\sigma}^2 \\
&= \mathrm{SIC}[S_{n-1}] - \left| \frac{1}{M} \sum_{m=1}^{M} y_m \exp(inx_m) \right|^2 \\
&\quad - \left| \frac{1}{M} \sum_{m=1}^{M} y_m \exp(-inx_m) \right|^2 + \frac{4\hat{\sigma}^2}{M},
\end{aligned} \tag{6.64}$$

which implies Eq.(6.32). ■

Chapter 7

Conclusions and future work

7.1 Conclusions

This dissertation was devoted to clarifying the mechanism of acquiring the generalization capability. We focused on the supervised learning scenario, and developed a theory of model selection and active learning. Our main concern was developing a theory valid for small sample cases, which can not be dealt with by most of the supervised learning theories (e.g. Mallows [72][73], Akaike [1], Takeuchi [133], Schwarz [115], Rissanen [99][100][101], Craven & Wahba [28], Murata *et al.* [82], Cohn [25], Cohn *et al.* [27], Konishi & Kitagawa [64], Fukumizu [38], Ishiguro *et al.* [55]).

In Chapter 4, the problem of model selection was discussed. We proposed a model selection criterion called the *subspace information criterion (SIC)*, which gives an unbiased estimate of the generalization error. Properties of SIC was investigated in various aspects including the comparison with a large number of existing model selection techniques. Computer simulations demonstrated that SIC works as well as existing methods in large sample cases, and it outperforms other methods in small sample cases.

In Chapter 5, the problem of active learning was discussed. We proposed batch and incremental active learning methods. The batch method can specify the optimal sampling locations for trigonometric polynomial models, while the incremental method can be applied to a wide range of models. Computer simulations showed that the proposed methods enable us to acquire higher levels of the generalization capability with a small number of training examples.

In Chapter 6, the problem of active learning with model selection, i.e., simultaneously optimizing sample points and models, was discussed. This subject was rather challenging since it can not be generally solved by simply combining existing active learning and

model selection techniques because of the active learning / model selection dilemma: The model should be fixed for active learning and conversely the sample points should be fixed for model selection. We gave a basic strategy for avoiding the dilemma, and a practical procedure of active learning with model selection for trigonometric polynomial models was proposed. Its excellent performance in small sample cases was demonstrated through computer simulations.

7.2 Problems for the future

In the final section of this dissertation, we show important subjects for the future.

7.2.1 Reference estimator framework for model selection

The main idea of SIC proposed in Chapter 4 was using an unbiased learning result function $\hat{f}_u$ for estimating the generalization error of $\hat{f}_\theta$. This idea can be interpreted as using another informative estimator for model selection (Tsuda *et al.* [137]). From this viewpoint, one expects that if another good estimate $\hat{f}_r$ of the learning target function f is available, the generalization error of $\hat{f}_\theta$ can be estimated more accurately. We call $\hat{f}_r$ a *reference estimator*. For substantiating the expectation, the following problem should be theoretically considered.

Problem 7.1 *Devise a method for estimating the generalization error of $\hat{f}_\theta$ by using another (maybe good) reference estimator $\hat{f}_r$.*

7.2.2 Variance of SIC

In Chapter 4, we derived SIC as an unbiased estimate of the generalization error. Since it is unbiased, the variance of SIC may not be small (see Shimodaira [120] for the variance of AIC). If the variance of SIC is drastically reduced by introducing a small bias in SIC, the model selection property will be further improved. To this end, the above reference estimator framework will be helpful: Instead of using an unbiased estimator $\hat{f}_u$, a slightly biased estimator $\hat{f}_\lambda$ is used as a reference estimator, where λ controls the bias of $\hat{f}_\lambda$. This problem is formulated, for example, as follows.

Problem 7.2 *Determine λ so that the following criterion is minimized:*

$$\mathrm{E}_{\boldsymbol{\epsilon}} \left(\mathrm{SIC}(\lambda) - J_G \right)^2 , \tag{7.1}$$

where E_ϵ *denotes the expectation over the noise and* J_G *denotes the generalization error.*

7.2.3 Model comparison

SIC gives a direct estimate of the generalization error and it is used for model selection. On the other hand, model selection can be performed without really estimating the generalization error. The difference of the generalization error of two models is only required:

$$J_G[\theta_1] - J_G[\theta_2]. \tag{7.2}$$

This is called *model comparison*. Estimating Eq.(7.2) is generally less difficult than estimating the generalization error itself. Therefore, solving the following problem is promising for improving the model selection performance.

Problem 7.3 *Devise a method for model comparison.*

7.2.4 Active learning for a set of models

In Chapter 6, the problem of active learning with model selection was discussed, and a basic strategy for solving the problem was given: Find a set of sample points which is commonly optimal to all model candidates. As we have shown, such a good set of sample points actually exists for trigonometric polynomial models. However, it may not exist in general.

In such cases, it is important to find a set of sample points which is *better* for all model candidates. Indeed, this gives a new direction of the active learning research. So far, theories of active learning have been developed for a fixed model. In practice, however, the model fixed in advance is often inappropriate so one wants to change the model. Then the sample points selected by active learning tend to be worse for the new model because the sample points are specially designed for the former model (Figure 7.1). A possible measure is to determine the sample points so that they are better for all model candidates. Then one can reduce the risk of changing models, which will make active learning techniques more practical. This problem is formulated as follows.

Problem 7.4 *Find a set* $\hat{\mathcal{X}}$ *of sample points such that*

$$\hat{\mathcal{X}} = \operatorname*{argmin}_{\mathcal{X}} \sum_{\theta \in \mathcal{M}} w(\theta) J_G[\mathcal{X}, \theta], \tag{7.3}$$

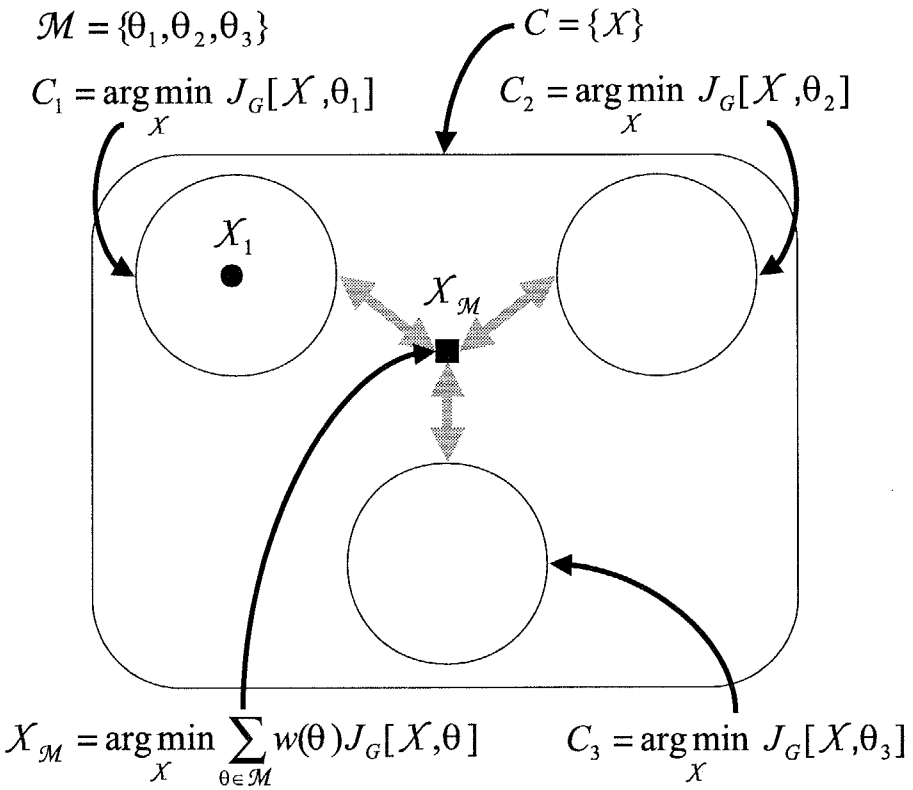


Figure 7.1: Let the set $\mathcal{M}$ of models be $\{\theta_1, \theta_2, \theta_3\}$. Suppose the current model is θ_1 and $\mathcal{X}_1$ is an optimal set of sample points for the model θ_1. If one changes the model to θ_2, then $\mathcal{X}_1$ may not be good sample points since it is specially designed for θ_1. On the other hand, if a set of sample points is determined so that it is better for all model candidates in $\mathcal{M}$ (denoted by $\mathcal{X}_{\mathcal{M}}$), then one can reduce the risk of changing models.

where θ is a model and $\mathcal{M}$ is a set of model candidates. $w(\theta)$ is some weight function and J_G denotes the generalization error.

7.2.5 Incremental active learning with model selection

Active learning with model selection discussed in Chapter 6 was in a batch manner. It will be useful if sample points and models are selected incrementally (Figure 7.2). In this process, there are two purposes for active learning (see MacKay [68]).

- Select an additional sample point so that it minimizes the generalization error for the current model.
- Select an additional sample point so that it is the most informative to select the model.

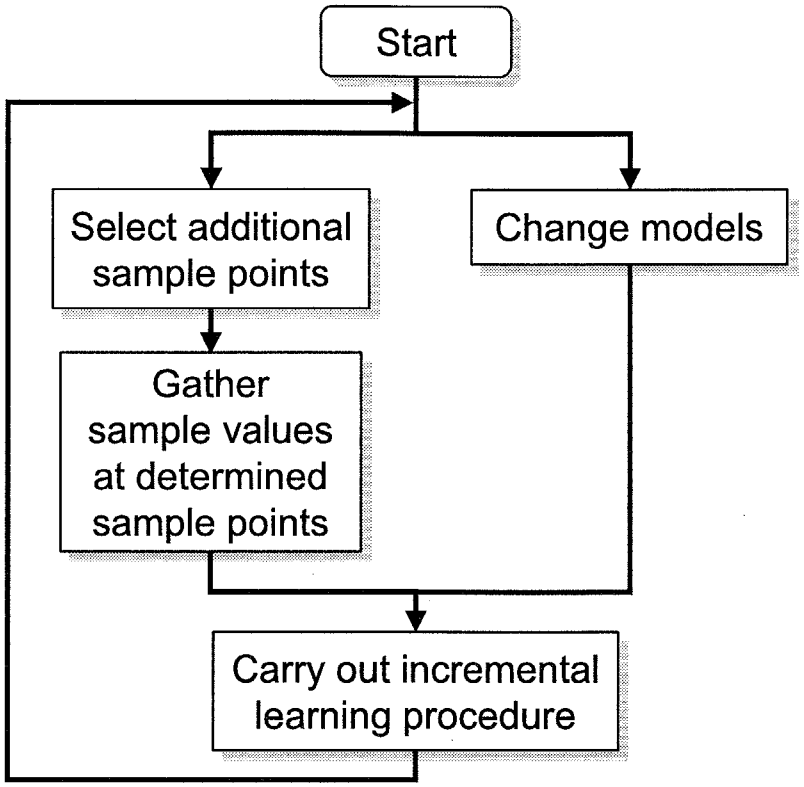


Figure 7.2: Incremental active learning with model selection.

This gives the following problem.

Problem 7.5 *Devise an incremental method for active learning with model selection.*

7.2.6 Infinite dimensional models

In this dissertation, we mainly consider a finite dimensional functional Hilbert space H to which the learning target function $f(\boldsymbol{x})$ belongs. It will be more flexible if an infinite dimensional H can be dealt with.

Problem 7.6 *Extend the theory of supervised learning developed in this dissertation so that it is applicable to an infinite dimensional functional Hilbert space H.*

7.2.7 Non-linear learning

In this dissertation, the learning result function $\hat{f}$ was assumed to be obtained by using a linear operator X as

$$\hat{f} = X\boldsymbol{y}, \tag{7.4}$$

where $\boldsymbol{y} = (y_1, y_2, \ldots, y_M)^\top$. This assumption means that the learning method is e.g. least mean squares learning or regularization learning with quadratic regularizers, and the regression model is linear:

$$\hat{f}(\boldsymbol{x}) = \sum_p w_p \varphi_p(\boldsymbol{x}), \tag{7.5}$$

where $\varphi_p(\boldsymbol{x})$ is a prefixed basis function and w_p is its coefficient.

Recently, sparsity inducing learning methods (i.e. most of the coefficients vanish) are rather popular (e.g. Bennett & Mangasarian [15], Vapnik [140][141], Williams [147], Mangasarian [74], Bradley *et al.* [20], Graepel *et al.* [44], Smola *et al.* [122], Smola & Schölkopf [123], Tipping [136], Tsuda *et al.* [137], Müller *et al.* [79]). They are practically very useful since necessary basis functions are automatically selected and the sparse solution saves the computational cost. In these learning methods, however, the operator X in Eq.(7.4) is no longer linear even if we are concerned with linear regression models. To cope with this situation, the following problem should be considered.

Problem 7.7 *Generalize the theory of supervised learning developed in this dissertation so that it can deal with a non-linear operator X.*

7.2.8 Non-linear regression models

Non-linear regression models such as multi-layer perceptrons are often preferred since their approximation ability is shown to outperform that of linear models (Jones [58], Girosi & Anzellotti [43], Barron [13], Girosi [41], Murata [80]). However, dealing with such non-linear models is very difficult because they do not generally satisfy the regularity condition for the asymptotic normality of the maximum likelihood estimator (Hagiwara *et al.* [47], Fukumizu [37]). Recently, Watanabe [146] rigorously showed the asymptotic generalization error for such non-linear models with singularities in the Bayesian ensemble learning case. A challenging future topic in this line is as follows.

Problem 7.8 *Extend the theory of supervised learning developed in this dissertation so that it is applicable to non-linear models.*

7.2.9 Minimum SIC learning

We used least mean squares learning and regularization learning in this dissertation. The role of learning methods is to determine a learning result function $\hat{f}$ from training examples

(Precisely, an operator X which gives $\hat{f}$ is determined). In the least mean squares learning case, a set $\{X_S\}$ of operators is prepared, and then the one that minimizes SIC is selected from the set. This can be interpreted as finding the minimizer of SIC with respect to X by multi-point search from candidates $\{X_S\}$, i.e., the minimizer among a finite number of candidates $\{X_S\}$ is selected. In the regularization learning case, the minimum of SIC with respect to X (precisely, the regularization parameter α) is analytically obtained. This corresponds to finding the minimizer of SIC under the constraint that X is a regularization operator.

From this point of view, the role of the learning methods is just a constraint for minimizing SIC. Therefore, it is very important to investigate whether the minimum of SIC with respect to X exists or not. If it exists, then one can obtain the best $\hat{X}$ that minimizes SIC without any constraint. This $\hat{X}$ is expected to give the optimal generalization capability. Otherwise, conditions for the existence of the minimum of SIC should be examined. Then one can obtain the best $\hat{X}$ that minimizes SIC under the existence conditions. The obtained $\hat{X}$ is expected to give a good generalization capability. Moreover, the existence conditions may lead to a new learning method since they are constraints for minimizing SIC. The above idea is summarized as follows.

Problem 7.9 *Investigate the existence of the minimum of SIC with respect to X.*

7.2.10 Supervised learning for points-of-interest estimation

The purpose of supervised learning was to estimate an underlying function $f(\boldsymbol{x})$ from a finite number of training examples. If $f(\boldsymbol{x})$ is successfully acquired, then one can estimate a future output value v_0 corresponding to a future input point $\boldsymbol{u}_0$ by $\hat{v}_0 = \hat{f}(\boldsymbol{u}_0)$. Since the main concern of this dissertation was investigating the mechanism of acquiring the generalization capability, a theory for estimating $f(\boldsymbol{x})$ was developed. However, if the purpose is just estimating an output value v_0 at a future input point $\boldsymbol{u}_0$, we do not have to estimate the entire function $f(\boldsymbol{x})$. It is enough to estimate the output value $v_0 = f(\boldsymbol{u}_0)$ only.

From the viewpoint of predicting output values of a function, methods for estimating the entire function f can be interpreted as using an estimate $\hat{f}$ of the entire function f for predicting all output values corresponding to all input points. However, if a future input point (or a test point) $\boldsymbol{u}_0$ is known in advance, direct estimation of the output value

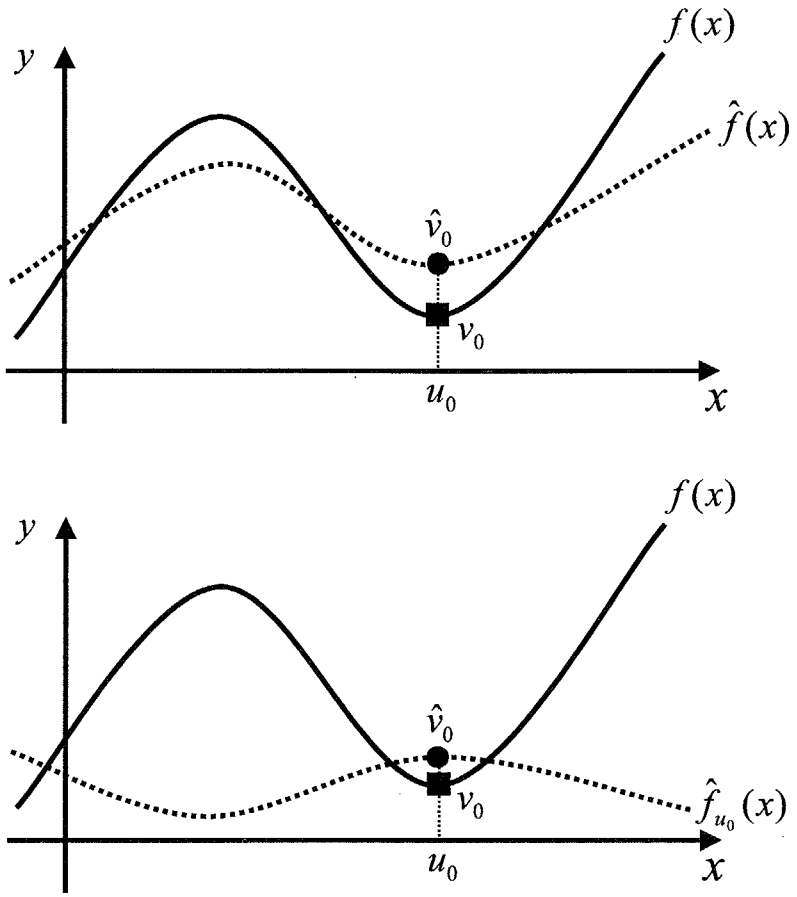


Figure 7.3: Points-of-interest estimation. $\hat{f}$ is better than $\hat{f}_{\boldsymbol{u}_0}$ as an estimate of the entire function f. However, $\hat{f}_{\boldsymbol{u}_0}$ is better than $\hat{f}$ as an estimate of the output value $v_0 = f(\boldsymbol{u}_0)$.

corresponding $\boldsymbol{u}_0$ is expected to result in better estimation than estimating the entire function (Figure 7.3). Research in this line can be found in many articles (e.g. Vapnik [139], Barron [12], Satoh [108][109], Shimodaira [119], Chapelle *et al.* [22]). In fact, this idea, which we call *points-of-interest estimation*, can also be applied to the case when the future test points are unknown: One do not have to obtain an estimate $\hat{f}$ of the entire function f *before* future input points are given. One can estimate an output value $v_0 = f(\boldsymbol{u}_0)$ *after* the future input point $\boldsymbol{u}_0$ is given, i.e., an output value is estimated from training examples every time a new input point is given. Clearly, this points-of-interest estimation is computationally very expensive. However, recent dramatic improvement of computation power is expected to enable us to perform the points-of-interest estimation in the near future. Therefore, solving the following problem is challenging and promising as future work.

Problem 7.10 *Develop a theory for estimating an output value of a function at one specified input point.*

7.2.11 Classification

The generalization measure J_G adopted in this dissertation was typically expressed as the expected squared distance between a learning result function $\hat{f}(\cdot)$ and the learning target function $f(\cdot)$ over the noise:

$$J_G = \mathrm{E}_{\epsilon} \int \left| \hat{f}(\boldsymbol{u}) - f(\boldsymbol{u}) \right|^2 w(\boldsymbol{u}) d\boldsymbol{u}, \tag{7.6}$$

where E_{ϵ} denotes the expectation over the noise and $w(\cdot)$ is some weight function. This generalization measure is natural in the regression cases where output values of the learning target function are estimated. On the other hand, in the classification cases where the class to which a sample belongs is estimated, the sign of the output values is important. For example, a sample $\boldsymbol{u}$ is assigned to the class $+1$ if $\mathrm{sgn}(f(\boldsymbol{u})) = 1$ and to the class -1 otherwise, where $\mathrm{sgn}(t) = 1$ if $t \geq 0$ and $\mathrm{sgn}(t) = -1$ if $t < 0$. In such a case, the following non-squared generalization measure is rather natural:

$$J_G = \mathrm{E}_{\epsilon} \int \left(1 - \mathrm{sgn}(\hat{f}(\boldsymbol{u}) f(\boldsymbol{u}))\right) w(\boldsymbol{u}) d\boldsymbol{u}. \tag{7.7}$$

Then we have the following future work.

Problem 7.11 *Extend the theory of supervised learning developed in this dissertation so that it is applicable to non-squared generalization measures.*

7.2.12 Unsupervised learning

Although we focused on supervised learning in this dissertation, studies on unsupervised learning is also an important topic in learning. Model selection, which we discussed as a subject of supervised learning, plays an essential role even in the unsupervised learning scenario, e.g. density estimation (Vapnik [140][141]) and principal component analysis (PCA) (Jolliffe [57]).

In the density estimation case, the distribution of input points is estimated by using a parametric model, where determining the structure of the parametric model is crucial for avoiding the *overfit* or *underfit*. PCA, which is a technique to find some important components of the distribution, is used for denoising, feature extraction, and compression

(Schölkopf *et al.* [113]). Denoising is performed by eliminating a certain number of minor components from the original data, expecting that the target signal in the data is concentrated in some leading components. Here, model selection plays an important role in balancing noise reduction with signal loss. Moreover, in non-linear PCA such as kernel PCA (Schölkopf *et al.* [113]), the type of non-linearity should also be determined. Note that image restoration from noisy images can also be formulated similarly (Sugiyama *et al.* [128]). Important future work in this line is as follows.

Problem 7.12 *Generalize the theory of supervised learning developed in this dissertation so that it is applicable to unsupervised learning.*

Acknowledgement

I am deeply indebted to Professor Hidemitsu Ogawa for his supervision since 1997, and providing me an excellent working environment. His valuable assistance and heart-warming encouragement in spite of his extremely busy schedule was very helpful. I would like to express my gratitude to Dr. Akira Hirabayashi (he is now with Yamaguchi University), Dr. Sethu Vijayakumar (he is now with University of Southern California), Dr. Akiko Nakashima (she is now with Toshiba Corporation), Professor Itsuo Kumazawa, and Professor Yukihiko Yamashita for their encouragement and valuable comments during my four year stay in Professor Ogawa's laboratory. I would like to thank Ms. Kyoko Satoh and all the members of Professor Ogawa's, Professor Kumazawa's, and Professor Yamashita's laboratories for their kindness and help.

For investigating the properties of SIC, discussion with Dr. Hironori Fujisawa of Tokyo Institute of Technology was very helpful. He also invited me to the Young Statisticians' Group Summer Seminar 2000. At the seminar, I had fruitful discussions with Dr. Katsuyuki Hagiwara of Mie University, Dr. Ken-ichi Satoh of Hiroshima University, and Dr. Seiya Imoto of Kyusyu University. I wish to express my gratitude to Dr. Shun-ichi Amari for reading my model selection paper and giving me a lot of valuable comments from the statistical point of view. I would like to thank Professor Ritei Shibata of Keio University for his comments when I visited his laboratory. I am very grateful to Dr. Klaus-Robert Müller for inviting me to the GMD FIRST in Germany as a visiting scientist. The joint project with him and Dr. Koji Tsuda was extremely fruitful. My gratitude also goes out to all the members of the Intelligent Data Analysis Group at the GMD FIRST for their warm hospitality.

My research project was financially supported by the Academic Encouragement Award from the Institute of Electronics, Information and Communication Engineers (IEICE) in 2000 and the Inose Academic Encouragement Award from the Incorporated Foundation of Electrical, Electronics, and Information Science Development in 1997.

Last but not least, I would like to thank my parents for giving me the opportunity to study for a Ph. D at Tokyo Institute of Technology.

Bibliography

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

[2] H. Akaike. Likelihood and the Bayes procedure. In N. J. Bernardo, M. H. DeGroot, D. V. Lindley, and A. F. M. Smith, editors, *Bayesian Statistics*, pages 141–166, Valencia, 1980. University Press.

[3] H. Akaike and G. Kitagawa, editors. *The Practice of Time Series Analysis I.* Asakura Syoten, Tokyo, Japan, 1994. (In Japanese).

[4] H. Akaike and G. Kitagawa, editors. *The Practice of Time Series Analysis II.* Asakura Syoten, Tokyo, Japan, 1995. (In Japanese).

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

[6] D. Allen. The relationship between variable selection and data augmentation and a method for prediction. *Technometrics*, 16:125–127, 1974.

[7] S. Amari. Theory of adaptive pattern classifiers. *IEEE Transactions on Electronic Computers*, EC-16(3):299–307, 1967.

[8] S. Amari, editor. *Recent Development in Neural Networks.* Saiensu-sya, 1993. (In Japanese).

[9] S. Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, 1998.

[10] S. Amari, N. Murata, K.-R. Müller, M. Finke, and H. H. Yang. Asymptotic statistical theory of overtraining and cross-validation. *IEEE Transactions on Neural Networks*, 8(5):985–996, 1997.

[11] N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68:337–404, 1950.

[12] A. R. Barron. Predicted squared error: A criterion for automatic model selection. In S. J. Farlow, editor, *Self-Organizing Methods in Modeling*, chapter 4, pages 87–103. Marcel Dekker, Inc., New York and Basel, 1984.

[13] A. R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, 39(3):930–945, 1993.

[14] A. Ben-Israel and T. N. E. Greville. *Generalized Inverses: Theory and Applications.* John Wiley & Sons, New York, 1974.

[15] K. P. Bennett and O. L. Mangasarian. Robust linear programming discrimination of two linearly inseparable sets. *Optimization Methods and Software*, 1:23–34, 1992.

[16] S. Bergman. *The Kernel Function and Conformal Mapping.* American Mathematical Society, Providence, Rhode Island, 1950.

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

[18] G. E. P. Box and W. G. Hunter. Sequential design of experiments for nonlinear models. In *Proceedings of IBM Scientific Computing Symposium in Statistics*, pages 113–137, 1965.

[19] H. Bozdogan, editor. *Proceedings of the First US/Japan Conference on the Frontiers of Statistical Modeling: An Informational Approach.* Kluwer Academic Publishers, Netherlands, 1994.

[20] P. S. Bradley, O. L. Mangasarian, and W. N. Street. Feature selection via mathematical programming. *INFORMS Journal on Computing*, 10:209–217, 1998.

[21] J. E. Cavanaugh and R. H. Shumway. A bootstrap variant of AIC for state space model selection. *Statistica Sinica*, 7:473–496, 1997.

[22] O. Chapelle, V. N. Vapnik, and J. Weston. Transductive inference for estimating values of functions. In S. A. Solla, T. K. Leen, and K.-R. Müller, editors, *Advances in Neural Information Processing Systems 12*, pages 421–427. MIT Press, 2000.

[23] V. Cherkassky, X. Shao, F. M. Mulier, and V. N. Vapnik. Model complexity control for regression using VC generalization bounds. *IEEE Transactions on Neural Networks*, 10(5):1075–1089, 1999.

[24] D. Cohn, L. Atlas, and R. Ladner. Improving generalization with active learning. *Machine Learning*, 15:201, 1994.

[25] D. A. Cohn. Neural network exploration using optimal experiment design. *Neural Networks*, 9(6):1071–1083, 1996.

[26] D. A. Cohn. Minimizing statistical bias with queries. In M. C. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems 9*, pages 417–423. The MIT Press, 1997.

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

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

[29] I. Daubechies. *Ten Lectures on Wavelets.* Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, USA, 1992.

[30] A. C. Davison and D. V. Hinkley. *Bootstrap Methods and their Application.* Cambridge University Press, 1997.

[31] B. Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1):1–26, 1979.

[32] B. Efron. How biased is the apparent error rate of a prediction rule? *Journal of the American Statistical Association*, 81(394):461–470, 1986.

[33] B. Efron and R. J. Tibshirani. *An Introduction to the Bootstrap.* Chapman & Hall, New York, 1993.

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

[35] M. Frank and D. R. Larson. A module frame concept for Hilbert C^*-modules. In L. W. Baggett and D. R. Larson, editors, *Functional and Harmonic Analysis of Wavelets*, volume 247 of *Contemporary Mathematics*. American Mathematical Society, San Antonio, TX, 1999.

[36] Y. Fujikoshi and K. Satoh. Modified AIC and C_P in multivariate linear regression. *Biometrika*, 84:707–716, 1997.

[37] K. Fukumizu. A regularity condition of the information matrix of a multilayer perceptron network. *Neural Networks*, 9(5):871–879, 1996.

[38] K. Fukumizu. Statistical active learning in multilayer perceptrons. *IEEE Transactions on Neural Networks*, 11(1):17–21, 2000.

[39] K. Fukumizu and S. Watanabe. Optimal training data and predictive error of polynomial approximation. *IEICE Transactions*, J79-A(5):1100–1108, 1996. (In Japanese).

[40] S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4(1):1–58, 1992.

[41] F. Girosi. Regularization theory, radial basis functions and networks. In J. H. Friedman H. Wechsler V. Cherkassky, editor, *From Statistics to Neural Networks*, F, Computer and Systems Sciences. Springer-Verlag, 1993.

[42] F. Girosi. An equivalence between sparse approximation and support vector machines. *Neural Computation*, 10(6):1455–1480, 1998.

[43] F. Girosi and G. Anzellotti. Convergence rate of approximation by translates. Technical Report 1288, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1992.

[44] T. Graepel, R. Herbrich, B. Schölkopf, A. J. Smola, P. L. Bartlett, K.-R. Müller, K. Obermayer, and R. C. Williamson. Classification on proximity data with LP-machines. In D. Willshaw and A. Murray, editors, *Proceedings of ICANN'99, International Conference on Artificial Neural Networks*, volume 1, pages 304–309. IEE Press, 1999.

[45] C. W. Groetsch. *The Theory of Tikhonov Regularization for Fredholm Equations of the First Kind*, volume 105 of *Research Notes in Mathematics*. Pitman Advanced Publishing Program, Boston, 1984.

[46] K. Hagiwara and K. Kuno. Regularization learning and early stopping in linear networks. In *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks*, volume 4, pages 511–516, Como, Italy, July 24-27 2000.

[47] K. Hagiwara, N. Toda, and S. Usui. Nonuniqueness of connecting weights and AIC in multi-layered neural networks. *Transactions of IEICE*, J76-D-II(9):2058–2065, 1993. (In Japanese).

[48] J. Hertz, A. Krogh, and R. G. Palmer. *Introduction to the Theory of Neural Computation*. Addison-Wesley, Redwood City, CA, 1991.

[49] J. J. Hopfield and D. W. Tank. "Neural" computation of decisions in optimization problems. *Biological Cybernetics*, 52:141–152, 1985.

[50] J. J. Hopfield and D. W. Tank. Computing with neural circuits: A model. *Science*, 223:625–633, 1986.

[51] C. M. Hurvich, J. S. Simonoff, and C. L. Tsai. Smoothing parameter selection in nonparametric regression using an improved Akaike information criterion. *Journal of the Royal Statistical Society, Series B*, 60:271–293, 1998.

[52] C. M. Hurvich and C. L. Tsai. Regression and time series model selection in small samples. *Biometrika*, 76:297–307, 1989.

[53] C. M. Hurvich and C. L. Tsai. Bias of the corrected AIC criterion for under-fitted regression and time series models. *Biometrika*, 78:499–509, 1991.

[54] C. M. Hurvich and C. L. Tsai. A corrected Akaike information criterion for vector autoregressive model selection. *Journal of Time Series Analysis*, 14:271–279, 1993.

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

[56] H. Iwaki, H. Ogawa, and A. Hirabayashi. Optimally generalizing neural networks with ability to recover from stuck-at r faults. *IEICE Transactions*, J83-D-II(2):805–813, 2000. (In Japanese).

[57] I. T. Jolliffe. *Principal Component Analysis.* Springer-Verlag, New York, 1986.

[58] L. K. Jones. A simple lemma on greedy approximation in Hilbert space and convergence rate for projection pursuit regression and neural network training. *Annals of Statistics*, 20(1):608–613, 1992.

[59] M. Kawato. *Computational Theory of Brain.* Sangyo-tosyo, 1996. (In Japanese).

[60] J. Kiefer. Optimum experimental designs. *Journal of the Royal Statistical Society, Series B*, 21:272–304, 1959.

[61] J. Kiefer and J. Wolfowitz. The equivalence of two extremum problems. *Annals of Mathematical Statistics*, 32:298, 1960.

[62] G. Kitagawa and W. Gersch. *Smoothness Priors Analysis of Time Series.* Number 116 in Lecture Notes in Statistics. Springer-Verlag, New York, 1996.

[63] T. Kohonen. *Associative Memory — A System-Theoretical Approach.* Springer-Verlag, Berlin, Heidelberg, and New York, 1977.

[64] S. Konishi and G. Kitagawa. Generalized information criterion in model selection. *Biometrika*, 83:875–890, 1996.

[65] S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22:79–86, 1951.

[66] K. Kunisch and J. Zou. Iterative choices of regularization parameters in linear inverse problem. *Inverse Problem*, 14:1247–1264, 1998.

[67] K. Li. Asymptotic optimality of C_L and generalized cross-validation in ridge regression with application to spline smoothing. *The Annals of Statistics*, 14(3):1101–1112, 1986.

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

[69] D. J. C. MacKay. The evidence framework applied to classification networks. *Neural Computation*, 4(5):720–736, 1992.

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

[71] D. J. C. MacKay. A practical Bayesian framework for backpropagation networks. *Neural Computation*, 4(3):448–472, 1992.

[72] C. L. Mallows. Choosing a subset regression. *Presented at the Central Regional Meeting of the Institute of Mathematical Statistics*, 1964.

[73] C. L. Mallows. Some comments on C_P. *Technometrics*, 15(4):661–675, 1973.

[74] O. L. Mangasarian. Mathematical programming in data mining. *Data Mining and Knowledge Discovery*, 42(1):183–201, 1997.

[75] D. Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information.* Freeman, San Francisco, 1982.

[76] A. D. R. McQuarrie and C. L. Tsai. *Regression and Time Series Model Selection.* World Scientific, Singapore, New Jersey, 1998.

[77] V. A. Morozov. *Regularization Methods for Ill-Posed Problems.* CRC Press, Inc., Boca Raton, 1993.

[78] F. Mosteller and D. Wallace. Inference in an authorship problem. A comparative study of discrimination methods applied to the authorship of the disputed Federalist papers. *Journal of the American Statistical Association*, 58:275–309, 1963.

[79] K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf. An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, 2001. (to appear).

[80] N. Murata. An integral representation of functions using three-layered networks and their approximation bounds. *Neural Networks*, 9(6):947–956, 1996.

[81] N. Murata. Bias of estimators and regularization terms. In *Proceedings of 1998 Workshop on Information-Based Induction Sciences (IBIS'98)*, pages 87–94, Izu, Japan, July 11-12 1998.

[82] N. Murata, S. Yoshizawa, and S. Amari. Network information criterion — Determining the number of hidden units for an artificial neural network model. *IEEE Transactions on Neural Networks*, 5(6):865–872, 1994.

[83] A. Nakashima and H. Ogawa. How to design a regularization term for improving generalization. In *Proceedings of ICONIP'99, The 6th International Conference on Neural Information Processing*, volume 1, pages 222–227, Perth, Australia, Nov. 1999.

[84] S. Nakazawa and H. Ogawa. Optimal realization of optimally generalizing neural networks. Technical Report NC96-60, IEICE, 1996. (In Japanese).

[85] K. Noda, E. Miyaoka, and M. Itoh. On bias correction of the Akaike information criterion in linear models. *Communications in Statistics. Theory and Methods*, 25:1845–1857, 1996.

[86] H. Ogawa. A theory of pseudo biorthogonal bases. *Transactions of IECE Japan*, J64-D(7):555–562, July 1981. (In Japanese).

[87] H. Ogawa. A unified approach to generalized sampling theorems. In *Proceedings of ICASSP'86, IEEE-IECEJ-ASJ International Conference on Acoustics, Speech, and Signal Processing*, pages 1657–1660, Tokyo, Japan, Apr. 1986.

[88] H. Ogawa. Projection filter regularization of ill-conditioned problem. In *Proceedings of SPIE, 808, Inverse Problems in Optics*, pages 189–196, 1987.

[89] H. Ogawa. Inverse problem and neural networks. In *Proceedings of IEICE 2nd Karuizawa Workshop on Circuits and Systems*, pages 262–268, Karuizawa, Japan, May 24–25 1989. (In Japanese).

[90] H. Ogawa. Neural network learning, generalization and over-learning. In *Proceedings of the ICIIPS'92, International Conference on Intelligent Information Processing & System*, volume 2, pages 1–6, Beijing, China, Oct. 30–Nov. 1 1992.

[91] H. Ogawa. Theory of pseudo biorthogonal bases and its application. In *Research Institute for Mathematical Science, RIMS Kokyuroku, 1067, Reproducing Kernels and their Applications*, number 1067, pages 24–38, Oct. 1998.

[92] H. Ogawa and T. Iijima. A theory of pseudo orthogonal bases. *Transactions of IECE Japan*, J58-D(5):271–278, May 1975. (In Japanese).

[93] H. Ogawa and I. Kumazawa. Radon transform and analog coding. In G. T. Herman, A. K. Louis, and F. Natterer, editors, *Mathematical Methods in Tomography*, volume 1497 of *Lecture Notes in Mathematics*, pages 229–241. Springer-Verlag, 1991.

[94] H. Ogawa and E. Oja. Projection filter, Wiener filter, and Karhunen-Loève subspaces in digital image restoration. *Journal of Mathematical Analysis and Applications*, 114(1):37–51, 1986.

[95] E. Oja and H. Ogawa. Parametric projection filter for image and signal restoration. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-34(6):1643–1653, Dec. 1986.

[96] M. J. L. Orr. Introduction to radial basis function networks. Technical report, Center for Cognitive Science, University of Edinburgh, 1996. (available from http://www.anc.ed.ac.uk/~mjo/papers/intro.ps.gz).

[97] J. Platt. A resource-allocating network for function interpolation. *Neural Computation*, 3(2):213–225, 1991.

[98] T. Poggio and F. Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9):1481–1497, 1990.

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

[100] J. Rissanen. Stochastic complexity. *Journal of the Royal Statistical Society, Series B*, 49(3):223–239, 1987.

[101] J. Rissanen. Fisher information and stochastic complexity. *IEEE Transactions on Information Theory*, IT-42(1):40–47, 1996.

[102] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In D. E. Rumelhart, J. L McClelland, and PDP Research Group, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, chapter 8, pages 318–362. The MIT Press, Cambridge, MA, 1986.

[103] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

[104] S. Saitoh. *Theory of Reproducing Kernels and Its Applications*, volume 189 of *Pitsman Research Notes in Mathematics Series*. Longman Scientific & Technical, UK, 1988.

[105] S. Saitoh. *Integral Transforms, Reproducing Kernels and Their Applications*, volume 369 of *Pitsman Research Notes in Mathematics Series*. Longman, UK, 1997.

[106] Y. Sakamoto, M. Ishiguro, and G. Kitagawa, editors. *Information Statistics*. Kyoritsu Syuppan, Tokyo, Japan, 1983. (In Japanese).

[107] K. Sato, M. Kobayashi, and Y. Fujikoshi. Variable selection for the growth curve model. *Journal of Multivariate Analysis*, 60:277–292, 1997.

[108] K. Satoh. AIC-type model selection criterion for multivariate linear regression with a future experiment. *Journal of the Japanese Statistical Society*, 27(2):135–140, 1997.

[109] K. Satoh. Modification of AIC-type criterion in multivariate normal linear regression with a future experiment. *Hiroshima Mathematical Journal*, 30(1):29–53, 2000.

[110] L. J. Savage. *The Foundation of Statistics*. Wiley, New York, 1954.

[111] R. Schatten. *Norm Ideals of Completely Continuous Operators*. Springer-Verlag, Berlin, 1970.

[112] B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors. *Advances in Kernel Methods: Support Vector Machines*. The MIT Press, Cambridge, MA, 1998.

[113] B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

[114] D. Schuurmans. A new metric-based approach to model selection. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, pages 552–558, 1997.

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

[116] X. Shao, V. Cherkassky, and W. Li. Measuring the VC-dimension using optimized experimental design. *Neural Computation*, 12(8):1969–1986, 2000.

[117] R. Shibata. Statistical aspects of model selection. In J. C. Willems, editor, *From Data to Model*, pages 375–394. Springer-Verlag, New York, 1989.

[118] R. Shibata. Bootstrap estimate of Kullback-Leibler information for model selection. *Statistica Sinica*, 7:375–394, 1997.

[119] H. Shimodaira. Predictive inference under misspecification and its model selection. Research Memorandum 642, The Institute of Statistical Mathematics, Tokyo, Japan, 1997.

[120] H. Shimodaira. An application of multiple comparison techniques to model selection. *Annals of Institute of Statistical Mathematics*, 50(1):1–13, 1998.

[121] J. S. Simonoff. Three sides of smoothing: Categorical data smoothing, nonparametric regression, and density estimation. *International Statistical Review*, 66:137–156, 1998.

[122] A. J. Smola, O. L. Mangasarian, and B. Schölkopf. Sparse kernel feature analysis. Technical Report 99-04, University of Wisconsin, Data Mining Institute, Madison, 1999.

[123] A. J. Smola and B. Schölkopf. Sparse greedy matrix approximation for machine learning. In *Proceedings of ICML-2000, International Conference on Machine Learning*, pages 911–918, 2000.

[124] P. Sollich. Query construction, entropy and generalization in neural network models. *Physical Review E*, 49:4637–4651, 1994.

[125] M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society, Series B*, 36:111–147, 1974.

[126] M. Stone. An asymptotic equivalence of choice of model by cross-validation and Akaike's criterion. *Journal of the Royal Statistical Society, Series B*, 39:44–47, 1977.

[127] N. Sugiura. Further analysis of the data by Akaike's information criterion and the finite corrections. *Communications in Statistics. Theory and Methods*, 7(1):13–26, 1978.

[128] M. Sugiyama, D. Imaizumi, and H. Ogawa. Subspace information criterion for image restoration — Mean squared error estimator for linear filters. In *Proceedings of the 12th Scandinavian Conference on Image Analysis*, Bergen, Norway, June 11–14 2001. (to appear).

[129] M. Sugiyama and H. Ogawa. Incremental projection learning for optimal generalization. *Neural Networks*, 14(1):53–66, 2001.

[130] M. Sugiyama and H. Ogawa. Properties of incremental projection learning. *Neural Networks*, 14(1):67–78, 2001.

[131] G. Szegö. *Orthogonal Polynomials*. American Mathematical Society, Providence, Rhode Island, 1939.

[132] A. Takemura. *Modern Mathematical Statistics*. Sobunsya, Tokyo, Japan, 1991. (In Japanese).

[133] K. Takeuchi. Distribution of information statistics and validity criteria of models. *Mathematical Science*, (153):12–18, 1976. (In Japanese).

[134] K. Takeuchi. On the selection of statistical models by AIC. *Journal of the Society of Instrument and Control Engineering*, 22(5):445–453, 1983. (In Japanese).

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

[136] M. E. Tipping. The relevance vector machine. In S. A. Solla, T. K. Leen, and K.-R. Müller, editors, *Advances in Neural Information Processing Systems 12*, pages 652–658. MIT Press, 2000.

[137] K. Tsuda, M. Sugiyama, and K.-R. Müller. Subspace information criterion for non-quadratic regularizers — Model selection for sparse regressors. Technical Report 120, GMD FIRST, Berlin, Germany, 2000.

[138] V. Vapnik and O. Chapelle. Bounds on error expectation for support vector machines. *Neural Computation*, 12(9):2013–2036, 2000.

[139] V. N. Vapnik. *Estimation of Dependencies Based on Empirical Data*. Springer-Verlag, New York, 1982.

[140] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, Berlin, 1995.

[141] V. N. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, Inc., New York, 1998.

[142] S. Vijayakumar and H. Ogawa. RKHS based functional analysis for exact incremental learning. *Neurocomputing*, 29(1–3):85–113, 1998.

[143] S. Vijayakumar and H. Ogawa. Improving generalization ability through active learning. *IEICE Transactions on Information and Systems*, E82-D(2):480–487, Feb. 1999.

[144] S. Vijayakumar, M. Sugiyama, and H. Ogawa. Training data selection for optimal generalization with noise variance reduction in neural networks. In M. Marinaro and R. Tagliaferri, editors, *Neural Nets WIRN Vietri-98*, pages 153–166. Springer-Verlag, 1998.

[145] H. Wahba. *Spline Model for Observational Data*. Society for Industrial and Applied Mathematics, Philadelphia and Pennsylvania, 1990.

[146] S. Watanabe. Algebraic analysis for non-identifiable learning machines. *Neural Computation*, 2001. (to appear).

[147] P. M. Williams. Bayesian regularization and pruning using a Laplace prior. *Neural Computation*, 7(1):117–143, 1995.

[148] W. Wong. A note on the modified likelihood for density estimation. *Journal of the American Statistical Association*, 78(382):461–463, 1983.

[149] R. X. Yue and F. J. Hickernell. Robust designs for fitting linear models with misspecification. *Statistica Sinica*, 9:1053–1069, 1999.