IFAC-SYSID2003 Aug. 27, 2003

Functional Analytic Framework for Model Selection

Fraunhofer Institut Rechnerarchitektur und Softwaretechnik

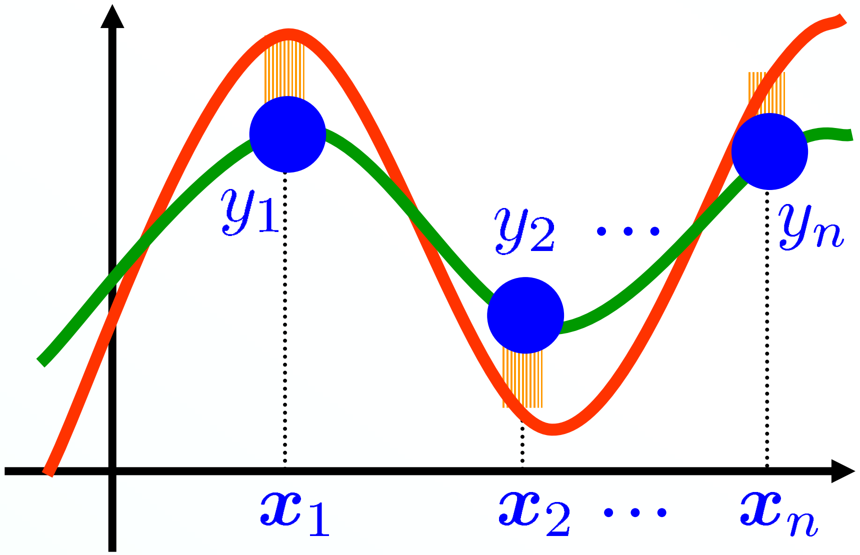
FIRST

Masashi Sugiyama

Tokyo Institute of Technology, Tokyo, Japan

Fraunhofer FIRST-IDA, Berlin, Germany

Regression Problem



$f(\boldsymbol{x})$:Underlying function

$\hat{f}(\boldsymbol{x})$:Learned function

$\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$:Training examples

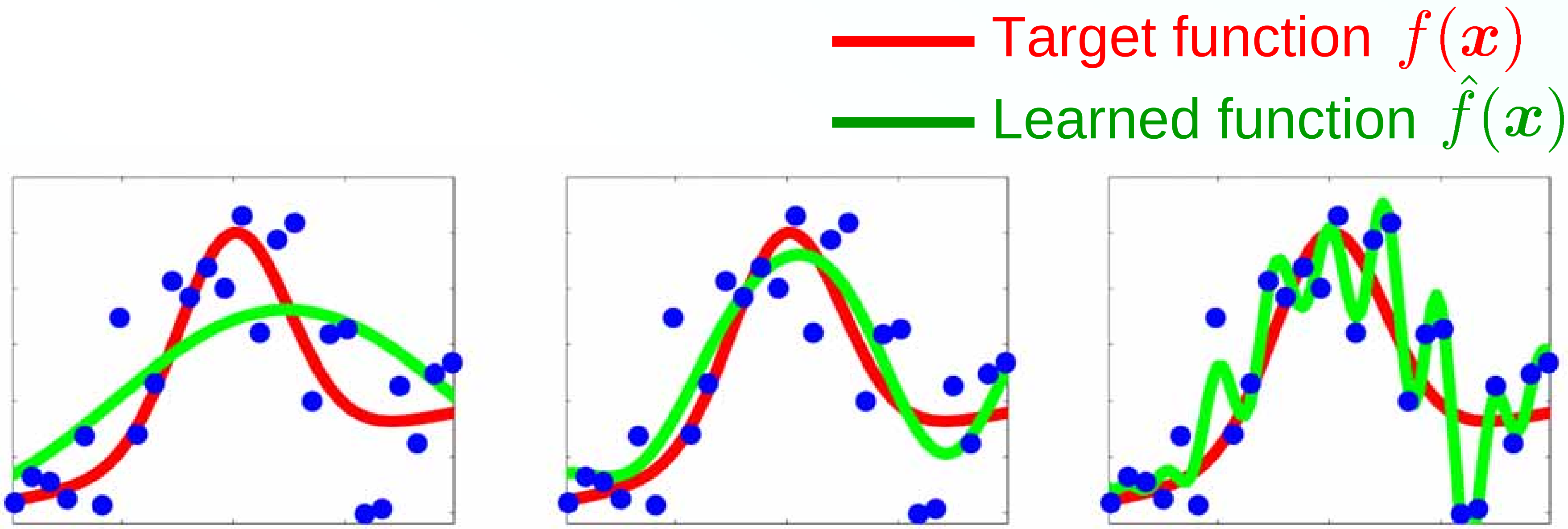
$$y_i = f(\boldsymbol{x}_i) + \epsilon_i$$

(noise)

$$\epsilon_i \overset{i.i.d.}{\sim} \text{mean } 0, \text{ variance } \sigma^2$$

From $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, obtain a good approximation $\hat{f}(\boldsymbol{x})$ to $f(\boldsymbol{x})$

Model Selection



Too simple Appropriate Too complex

Choice of the model is extremely important for obtaining good learned function $\hat{f}(\boldsymbol{x})$!

(Model refers to, e.g., regularization parameter)

Aims of Our Research

- Model is chosen such that a generalization error estimator is minimized.
- Therefore, model selection research is essentially to pursue an accurate estimator of the generalization error.
- We are interested in
  - Having a novel method in different framework.
  - Estimating the generalization error with small (finite) samples.

Formulating Regression Problem as Function Approximation Problem

- H : A functional Hilbert space
- We assume $f, \hat{f} \in H$
- We shall measure the "goodness" of the learned function $\hat{f}$ (or the generalization error) by

$$\mathrm{E}\|\hat{f} - f\|^2$$

E :Expectation over noise

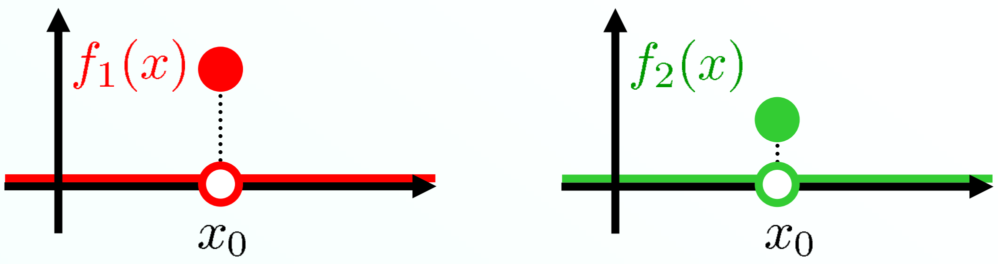
$\|\cdot\|$:Norm in H

Function Spaces for Learning

- In learning problems, we sample values of the target function at sample points (e.g., $f(\boldsymbol{x}_1)$).
- Therefore, values of the target function at sample points should be specified.
- This means that usual L_2 -space is not suitable for learning problems.

L_2 is spanned by

$$\left\{ f \;\middle|\; \int |f(x)|^2 dx < \infty \right\}$$



f_1 and f_2 have different values at x_0

$$f_1(x_0) \neq f_2(x_0)$$

But they are treated as the same function in L_2

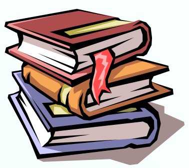
$$f_1 = f_2$$

Reproducing Kernel Hilbert Spaces

- In a reproducing kernel Hilbert space (RKHS), a value of a function at an input point is always specified.
- Indeed, an RKHS H has the reproducing kernel $K(\boldsymbol{x}, \boldsymbol{x}')$ with reproducing property:

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

$\langle \cdot, \cdot \rangle$:Inner product in H



Sampling Operator

- For any RKHS H, there exists a linear operator A from H to $\mathbb{R}^n$ such that

$$Af = (f(\boldsymbol{x}_1), f(\boldsymbol{x}_2), \ldots, f(\boldsymbol{x}_n))^\top$$

- Indeed, $A = \sum_{i=1}^{n} \left(\boldsymbol{e}_i \otimes \overline{K(\cdot, \boldsymbol{x}_i)} \right)$

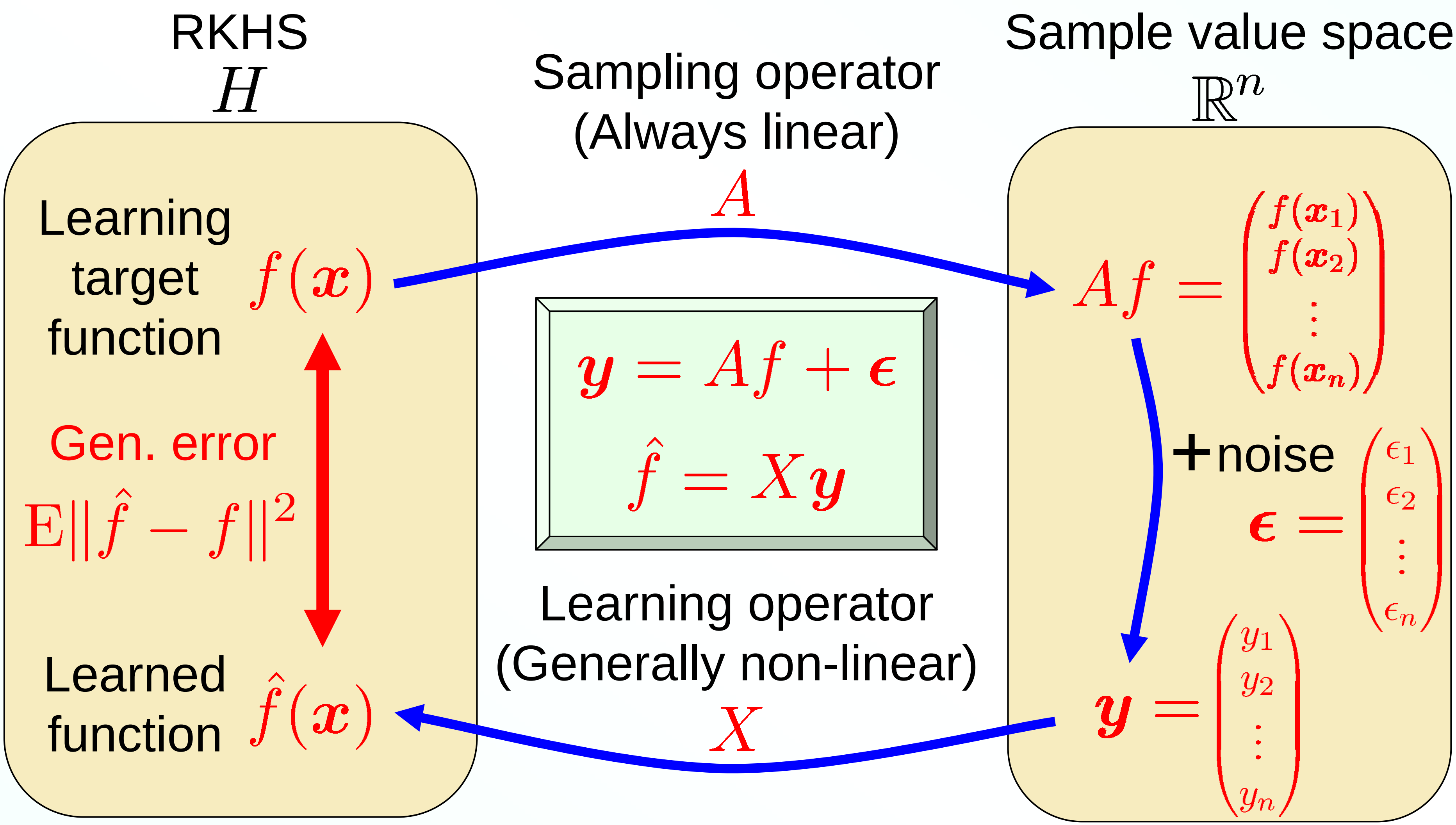
$(\cdot \otimes \bar{\cdot})$:Neumann-Schatten product

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

For vectors, $(f \otimes \overline{g}) = f g^\top$

$\boldsymbol{e}_i$: i-th standard basis in $\mathbb{R}^n$

Our Framework



E :Expectation over noise

Tricks for Estimating Generalization Error

- We want to estimate $\mathrm{E}\|\hat{f}-f\|^2$. But it includes unknown f so it is not straightforward.
- To cope with this problem,
  - We shall estimate only its essential part

$$\mathrm{E}\|\hat{f}-f\|^2 = \underbrace{\mathrm{E}\|\hat{f}\|^2 - 2\mathrm{E}\langle \hat{f}, f\rangle}_{\text{Essential part } J} + \underbrace{\|f\|^2}_{\text{Constant}}$$

$$J = \mathrm{E}\|\hat{f}-f\|^2 - \|f\|^2$$

  - We focus on the kernel regression model:

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

$K(\boldsymbol{x}, \boldsymbol{x}')$:Reproducing kernel of H

A Key Lemma

For the kernel regression model,
the essential gen. error J is expressed by

$$J = \mathrm{E}\left(\|\hat{f}\|^2 - 2\langle \hat{f}, A^{\dagger}\boldsymbol{y} \rangle + 2\langle \hat{f}, A^{\dagger}\boldsymbol{\epsilon} \rangle \right)$$

$J = \mathrm{E}\|\hat{f} - f\|^2 - \|f\|^2$

E :Expectation over noise

Unknown target function f can be erased!

$Af = (f(\boldsymbol{x}_1), f(\boldsymbol{x}_2), \ldots, f(\boldsymbol{x}_n))^\top$ $\quad \boldsymbol{y} = (y_1, y_2, \ldots, y_n)^\top$

$A^{\dagger}$:Generalized inverse $\quad \boldsymbol{\epsilon} = (\epsilon_1, \epsilon_2, \ldots, \epsilon_n)^\top$

Estimating Essential Part J

$$J = \mathrm{E}\left(\|\hat{f}\|^2 - 2\langle \hat{f}, A^\dagger \boldsymbol{y}\rangle + 2\langle \hat{f}, A^\dagger \boldsymbol{\epsilon}\rangle\right)$$

$$\boldsymbol{\epsilon} = (\epsilon_1, \epsilon_2, \ldots, \epsilon_n)^\top$$

- $\|\hat{f}\|^2 - 2\langle \hat{f}, A^\dagger \boldsymbol{y}\rangle + 2\langle \hat{f}, A^\dagger \boldsymbol{\epsilon}\rangle$ is an unbiased estimator of the essential gen. error J .
- However, the noise vector $\boldsymbol{\epsilon}$ is unknown.
- Let us define

$$preSIC = \|\hat{f}\|^2 - 2\langle \hat{f}, A^\dagger \boldsymbol{y}\rangle + 2\mathrm{E}\langle \hat{f}, A^\dagger \boldsymbol{\epsilon}\rangle$$

- Clearly, it is still unbiased: $\mathrm{E}[preSIC] = J$
- We would like to handle $\mathrm{E}\langle \hat{f}, A^\dagger \boldsymbol{\epsilon}\rangle$ well.

How to Deal with $\mathrm{E}\langle \hat{f}, A^\dagger \boldsymbol{\epsilon} \rangle$

$$preSIC = \|\hat{f}\|^2 - 2\langle \hat{f}, A^\dagger \boldsymbol{y} \rangle + 2\mathrm{E}\langle \hat{f}, A^\dagger \boldsymbol{\epsilon} \rangle$$

$$\hat{f} = X\boldsymbol{y} \qquad \boldsymbol{y} = (y_1, y_2, \ldots, y_n)^\top$$

Depending on the type of learning operator X we consider the following three cases.

A) X is linear.

B) X is non-linear but twice almost differentiable.

C) X is general non-linear.

A) Examples of Linear Learning Operator

- Kernel ridge regression
- A particular Gaussian process regression
- Least-squares support vector machine

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

α_i :Parameters to be learned

$$\min_{\{\alpha_i\}} \left[\sum_{i=1}^{n} \left(\hat{f}(\boldsymbol{x}_i) - y_i \right)^2 + \lambda \|\hat{f}\|^2 \right]$$

λ :Ridge parameter

A) Linear Learning

When the learning operator X is linear,

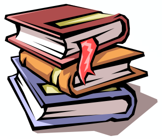
$$\mathrm{E}\langle \hat{f}, A^{\dagger}\boldsymbol{\epsilon}\rangle = \sigma^2 \mathrm{tr}\,(XX^*)$$

$\hat{f} = X\boldsymbol{y}$ X^*:Adjoint of X

$$preSIC = \|\hat{f}\|^2 - 2\langle \hat{f}, A^{\dagger}\boldsymbol{y}\rangle + 2\mathrm{E}\langle \hat{f}, A^{\dagger}\boldsymbol{\epsilon}\rangle$$

- This induces the subspace information criterion (SIC):

M. Sugiyama & H. Ogawa (Neural Comp, 2001)
M. Sugiyama & K.-R. Müller (JMLR, 2002)



$$SIC = \|\hat{f}\|^2 - 2\langle \hat{f}, A^{\dagger}\boldsymbol{y}\rangle + 2\sigma^2 \mathrm{tr}\,(XX^*)$$

- SIC is unbiased with finite samples:

$$\mathrm{E}[SIC] = J$$

How to Deal with $\mathrm{E}\langle \hat{f}, A^{\dagger}\boldsymbol{\epsilon}\rangle$

$$preSIC = \|\hat{f}\|^2 - 2\langle \hat{f}, A^{\dagger}\boldsymbol{y}\rangle + 2\mathrm{E}\langle \hat{f}, A^{\dagger}\boldsymbol{\epsilon}\rangle$$

$$\hat{f} = X\boldsymbol{y} \qquad \boldsymbol{y} = (y_1, y_2, \ldots, y_n)^{\top}$$

Depending on the type of learning operator X we consider the following three cases.

A) X is linear.

B) X is non-linear but twice almost differentiable.

C) X is general non-linear.

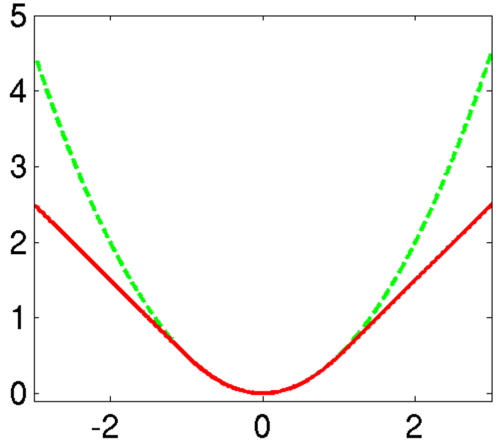
B) Examples of Twice Almost Differentiable Learning Operator

- Support vector regression with Huber's loss

$$\min_{\{\alpha_i\}} \left[\sum_{i=1}^{n} \rho\left(\hat{f}(\boldsymbol{x}_i) - y_i\right) + \lambda \|\hat{f}\|^2 \right]$$

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

λ :Ridge parameter



$$\rho(y) = \begin{cases} \frac{1}{2}y^2 & (|y| \leq t) \\ t|y| - \frac{1}{2}t^2 & (|y| > t) \end{cases}$$

t :Threshold

B) Twice Differentiable Learning

For the Gaussian noise, we have

$$\mathrm{E}\langle \hat{f}, A^{\dagger}\boldsymbol{\epsilon}\rangle = \mathrm{E}\left(\sigma^2 \sum_{i=1}^{n} \frac{\partial [A^{\dagger}X]_i(\boldsymbol{y})}{\partial y_i}\right)$$

$[A^{\dagger}X](\boldsymbol{y})$:Vector-valued function

$$preSIC = \|\hat{f}\|^2 - 2\langle \hat{f}, A^{\dagger}\boldsymbol{y}\rangle + 2\mathrm{E}\langle \hat{f}, A^{\dagger}\boldsymbol{\epsilon}\rangle$$

- SIC for twice almost differentiable learning:

$$SIC = \|\hat{f}\|^2 - 2\langle \hat{f}, A^{\dagger}\boldsymbol{y}\rangle + 2\sigma^2 \sum_{i=1}^{n} \frac{\partial [A^{\dagger}X]_i(\boldsymbol{y})}{\partial y_i}$$

- It reduces to the original SIC if X is linear.
- It is still unbiased with finite samples:

$$\mathrm{E}[SIC] = J$$

How to Deal with $\mathrm{E}\langle \hat{f}, A^{\dagger}\boldsymbol{\epsilon}\rangle$

$$preSIC = \|\hat{f}\|^2 - 2\langle \hat{f}, A^{\dagger}\boldsymbol{y}\rangle + 2\mathrm{E}\langle \hat{f}, A^{\dagger}\boldsymbol{\epsilon}\rangle$$

$$\hat{f} = X\boldsymbol{y} \qquad \boldsymbol{y} = (y_1, y_2, \ldots, y_n)^{\top}$$

Depending on the type of learning operator X we consider the following three cases.

A) X is linear.

B) X is non-linear but twice almost differentiable.

C) X is general non-linear.

C) Examples of General Non-Linear Learning Operator

- Kernel sparse regression

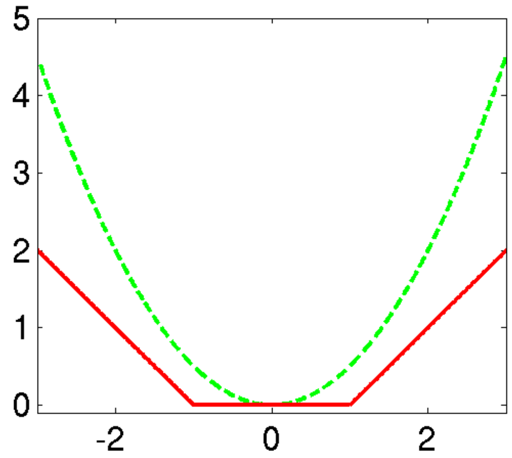
$$\hat{f}(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i K(\boldsymbol{x}, \boldsymbol{x}_i)$$

$$\min_{\{\alpha_i\}} \left[\sum_{i=1}^{n} \left(\hat{f}(\boldsymbol{x}_i) - y_i \right)^2 + \lambda \sum_{i=1}^{n} |\alpha_i| \right]$$

- Support vector regression with Vapnik's loss

$$\min_{\{\alpha_i\}} \left[\sum_{i=1}^{n} \left| \hat{f}(\boldsymbol{x}_i) - y_i \right|_{\varepsilon} + \lambda \|\hat{f}\|^2 \right]$$

$$|y|_{\varepsilon} = \begin{cases} 0 & (|y| \le \varepsilon) \\ |y| - \varepsilon & (|y| > \varepsilon) \end{cases}$$



C) General Non-Linear Learning

Approximation by the bootstrap

$$\mathrm{E}\langle \hat{f}, A^\dagger \boldsymbol{\epsilon} \rangle \approx \mathrm{E}^b \langle \hat{f}^b, A^\dagger \hat{\boldsymbol{\epsilon}}^b \rangle$$

E^b :Expectation over bootstrap replications

$$preSIC = \|\hat{f}\|^2 - 2\langle \hat{f}, A^\dagger \boldsymbol{y} \rangle + 2\mathrm{E}\langle \hat{f}, A^\dagger \boldsymbol{\epsilon} \rangle$$

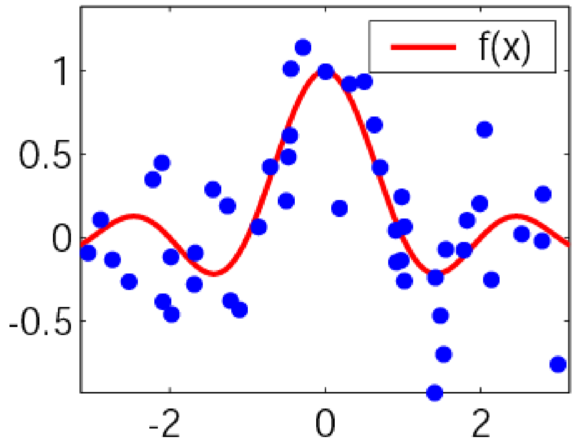
- Bootstrap approximation of SIC (BASIC):

$$BASIC = \|\hat{f}\|^2 - 2\langle \hat{f}, A^\dagger \boldsymbol{y} \rangle + 2\mathrm{E}^b \langle \hat{f}^b, A^\dagger \hat{\boldsymbol{\epsilon}}^b \rangle$$

- BASIC is almost unbiased:

$$\mathrm{E}[BASIC] \approx J$$

Simulation: Learning Sinc function

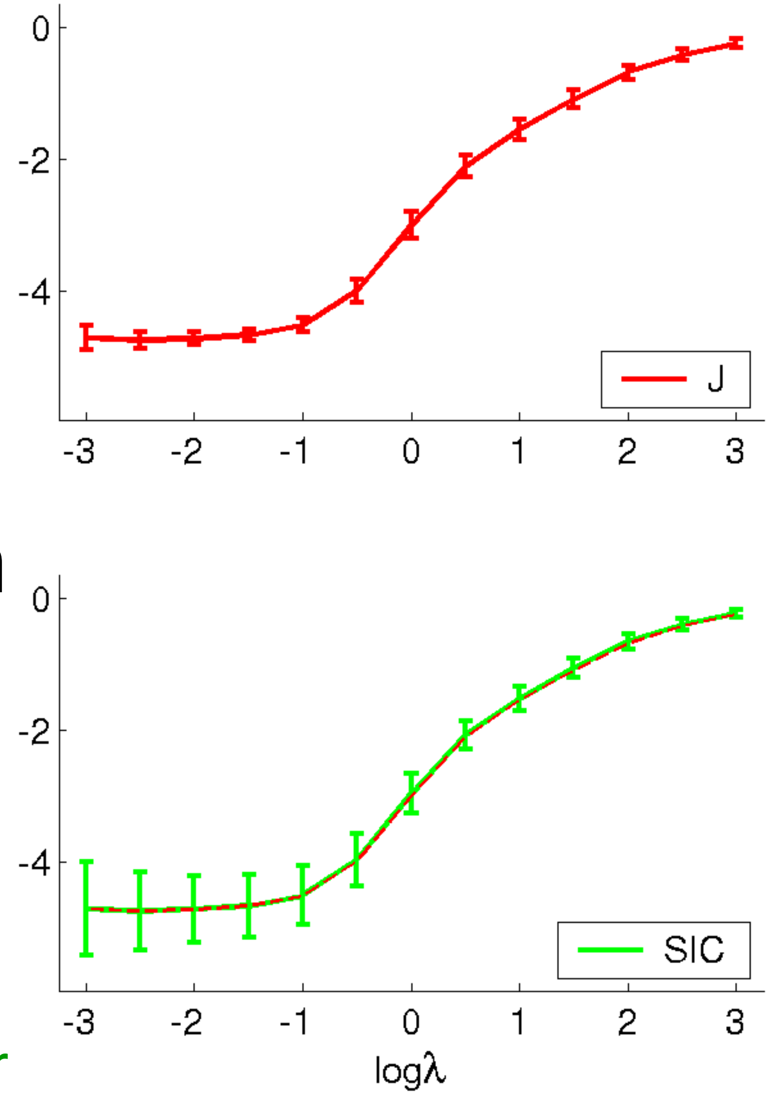


- H :Gaussian RKHS
- Kernel ridge regression

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

$$\min_{\{\alpha_i\}} \left[\sum_{i=1}^{n} \left(\hat{f}(\boldsymbol{x}_i) - y_i \right)^2 + \lambda \|\hat{f}\|^2 \right]$$

λ :Ridge parameter



Simulation: DELVE Data Sets

Normalized test error

Data	RSIC	Cross Validation	Empirical Bayes
Abalone	1.0144 ± 0.0002	1.0146 ± 0.0002	1.0204 ± 0.0003
Boston	1.0016 ± 0.0007	1.0071 ± 0.0007	1.1406 ± 0.0008
Bank-8fm	1.0703 ± 0.0001	1.0708 ± 0.0001	1.0030 ± 0.0001
Bank-8nm	1.0002 ± 0.0004	1.0461 ± 0.0005	1.0477 ± 0.0005
Bank-8fh	1.0025 ± 0.0003	1.0026 ± 0.0003	1.0003 ± 0.0003
Bank-8nh	1.0028 ± 0.0005	1.2177 ± 0.0008	1.4200 ± 0.0008
Kin-8fm	1.0000 ± 0.0001	1.0010 ± 0.0001	1.4548 ± 0.0004
Kin-8nm	1.0097 ± 0.0010	1.0241 ± 0.0007	1.0371 ± 0.0006
Kin-8fh	1.0021 ± 0.0003	1.0057 ± 0.0003	1.2025 ± 0.0001
Kin-8nh	1.0451 ± 0.0009	1.0017 ± 0.0004	1.0361 ± 0.0004

Red: Best or comparable (95%t-test)

Conclusions

- We provided a functional analytic framework for regression, where the generalization error is measured using the RKHS norm: $\mathrm{E}\|\hat{f} - f\|^2$
- Within this framework, we derived a generalization error estimator called SIC.

A) Linear learning (Kernel ridge, GPR, LS-SVM):
SIC is exact unbiased with finite samples.

B) Twice almost differentiable learning (SVR+Huber):
SIC is exact unbiased with finite samples.

C) Non-linear learning (K-sparse, SVR+Vapnik):
BASIC is almost unbiased.