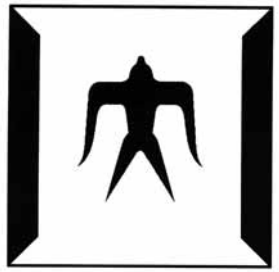
FIT2002　　2002年9月25日～28日

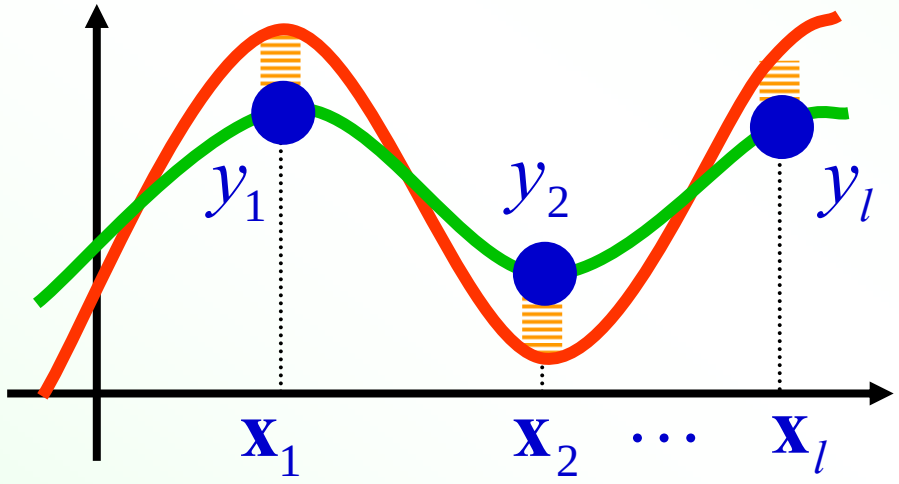
サポートベクター回帰のモデル選択



東京工業大学　計算工学専攻

杉山　将

教師付き学習 (回帰)



$f(\mathbf{x})$: 学習したい関数

$\hat{f}(\mathbf{x})$: 学習結果の関数

$\{\mathbf{x}_i, y_i\}_{i=1}^{l}$: 訓練データ

$$y_i = f(\mathbf{x}_i) + \varepsilon_i$$

$\varepsilon_i \overset{i.i.d.}{\sim}$ 平均 0, 分散 σ^2

訓練データ $\{\mathbf{x}_i, y_i\}_{i=1}^{l}$ を用いて、
未知の $f(\mathbf{x})$ にできるだけ近い $\hat{f}(\mathbf{x})$ を求めよ

サポートベクター回帰

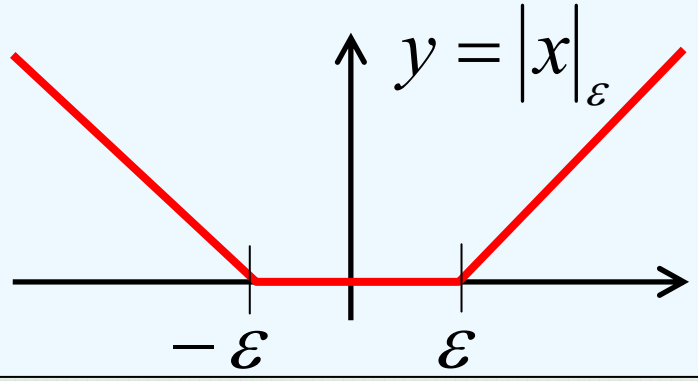
Vapnik (1995)

次式を最小にする $\hat{f}(x)$ を求めよ

$$\min_{\hat{f}\in H}\left[\frac{1}{2}\left\|\hat{f}\right\|^2+\frac{C}{l}\sum_{i=1}^{l}\left|\hat{f}(\mathbf{x}_i)-y_i\right|_\varepsilon\right]$$

“滑らかさ”　　訓練誤差

Vapnik の ε-insensitive loss function



H : 再生核ヒルベルト空間

$\|\cdot\|$: H のノルム

C : 正則化パラメータ

l : 訓練データ数

→ ロバスト回帰の一種

2次計画問題

$$\min_{\hat{f}\in H}\left[\frac{1}{2}\left\|\hat{f}\right\|^2+\frac{C}{l}\sum_{i=1}^{l}\left|\hat{f}(\mathbf{x}_i)-y_i\right|_{\varepsilon}\right]$$

$$\min_{\hat{f}\in H,\xi^+,\xi^-}\left[\frac{1}{2}\left\|\hat{f}\right\|^2+\frac{C}{l}\sum_{i=1}^{l}(\xi_i^+ +\xi_i^-)\right]$$

$$\text{subject to}\quad \hat{f}(\mathbf{x}_i)-y_i\le\varepsilon+\xi_i^+$$

$$y_i-\hat{f}(\mathbf{x}_i)\le\varepsilon+\xi_i^-$$

$$\xi_i^+,\xi_i^-\ge 0$$

必ず大域解が求まる！

2次計画問題 (双対)

Representer Theorem (Kimeldorf & Wahba, 1970)

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

$K(\mathbf{x}, \mathbf{x}') : H$ の再生核

$$\alpha_i = \alpha_i^+ - \alpha_i^-$$

$$\max_{\boldsymbol{\alpha}^+, \boldsymbol{\alpha}^-} \left[-\frac{1}{2} \sum_{i,j=1}^{l} (\alpha_i^+ - \alpha_i^-)(\alpha_j^+ - \alpha_j^-) K(\mathbf{x}_i, \mathbf{x}_j) - \varepsilon \sum_{i=1}^{l} (\alpha_i^+ + \alpha_i^-) + \sum_{i=1}^{l} y_i (\alpha_i^+ - \alpha_i^-) \right]$$

$$\text{subject to } \sum_{i=1}^{l} (\alpha_i^+ - \alpha_i^-) = 0$$

$$\alpha_i^+, \alpha_i^- \in [0, C/l]$$

サポートベクター回帰の特徴

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

- 解がスパースになる。即ち、

  多くの i に対して $\alpha_i = 0$ となる

- 大規模な問題を高速に解くことができる優れたソフトウェアが無料で入手できる。例えば、

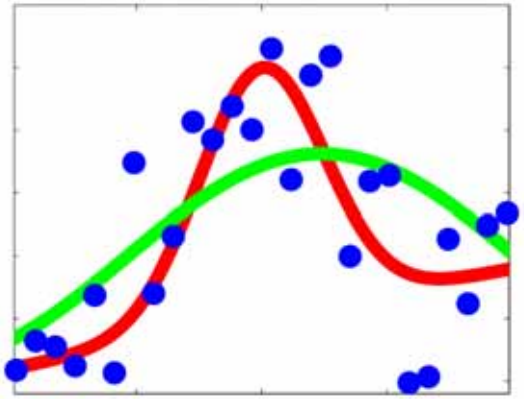
  SVM Light, SVM Torch,
  mySVM, LIBSVM 等

汎化能力と正則化パラメータ

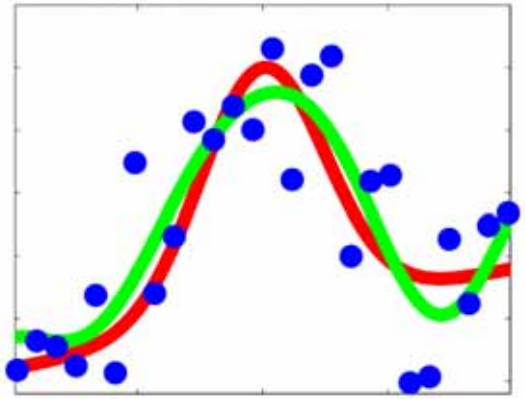
$$\min_{\hat{f}\in H}\left[\frac{1}{2}\left\|\hat{f}\right\|^2+\frac{C}{l}\sum_{i=1}^{l}\left|\hat{f}(\mathbf{x}_i)-y_i\right|_{\varepsilon}\right]$$

C : 正則化パラメータ

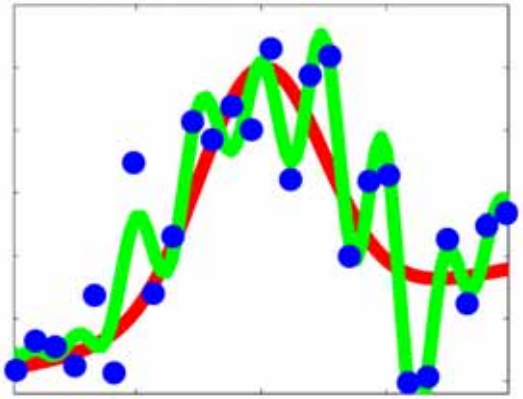
—— 学習したい関数
—— 学習結果の関数



C が小さすぎる



C が適切

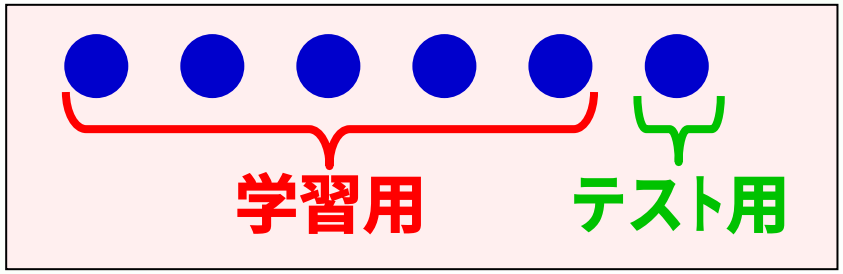


C が大きすぎる

C の値を適切に決定しなければ
高い汎化能力は得られない

クロスバリデーション

訓練データを k 群に分け、$(k-1)$ 群で学習し、
残りで汎化能力を推定する。これを全ての
k 種類の組み合わせに対して行なう



多くの場合にうまく働くが、万能ではない

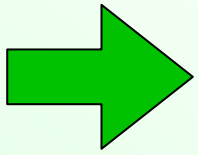
**モデル選択研究の更なる発展のためには、
異なった立場から研究を行うことが重要**

関数解析的な枠組み

- 学習したい真の関数 $f(\mathbf{x})$ が $K(\mathbf{x}, \mathbf{x}')$ を再生核とする**関数空間** H に含まれると仮定

- 関数空間内での自然な距離尺度である**ノルム**を用いて学習結果の関数 $\hat{f}(\mathbf{x})$ の"良さ"（関数の近似誤差）を評価する

$$\left\| \hat{f} - f \right\|^2 \qquad \|\cdot\| : H \text{ のノルム}$$



$\left\| \hat{f} - f \right\|^2$ **を推定したい！**

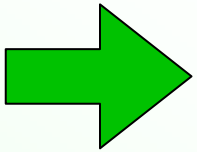
Subspace Information Criterion

Sugiyama & Ogawa (2001), Sugiyama & Mueller (2002)

- SICは有限の訓練データに対して

$$E\,\mathrm{SIC} = E\left\|\hat{f} - f\right\|^2 - \text{定数}$$

E：雑音に関する平均



しかし、これまでSICは線形な学習法にしか適用できなかった

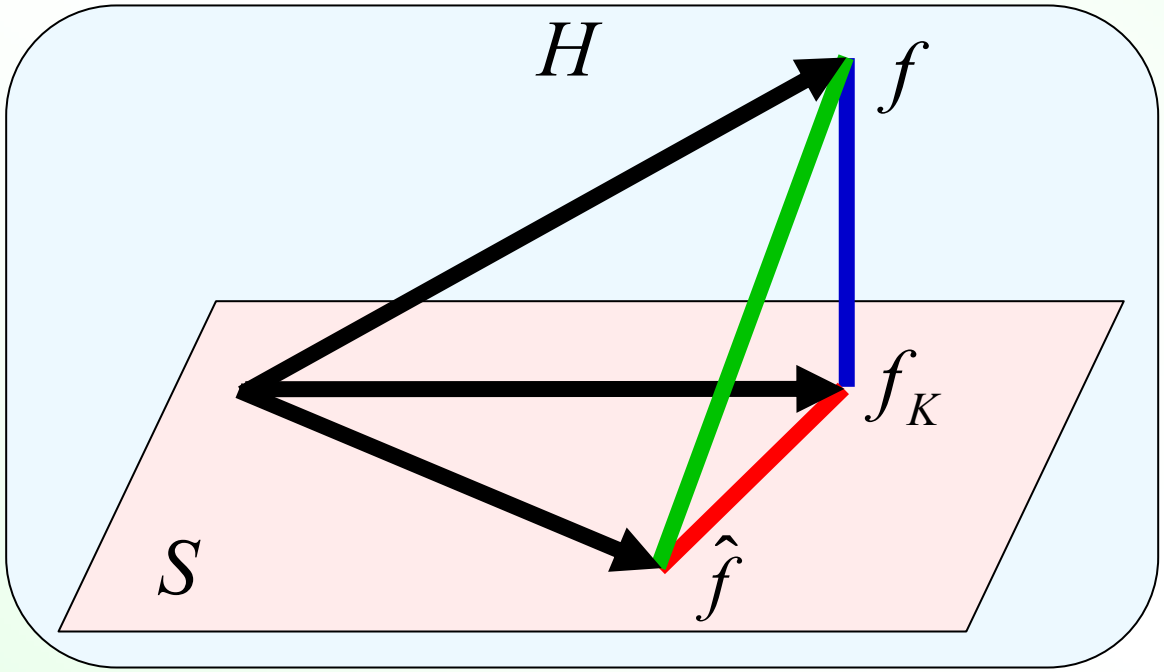
（例：最小二乗推定、リッジ推定）

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i K(\mathbf{x}, \mathbf{x}_i) \qquad \alpha_i = \sum_{j=1}^{l} c_{i,j} y_j$$

サポートベクター回帰などの非線形な学習法にも適用できるようにSICを拡張する

SICのアイデア (1)

$$\underbrace{\left\|\hat{f}-f\right\|^2}_{\text{汎化誤差}}=\underbrace{\left\|\hat{f}-f_K\right\|^2}_{\text{本質的}}+\underbrace{\left\|f_K-f\right\|^2}_{\text{定数}}$$



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

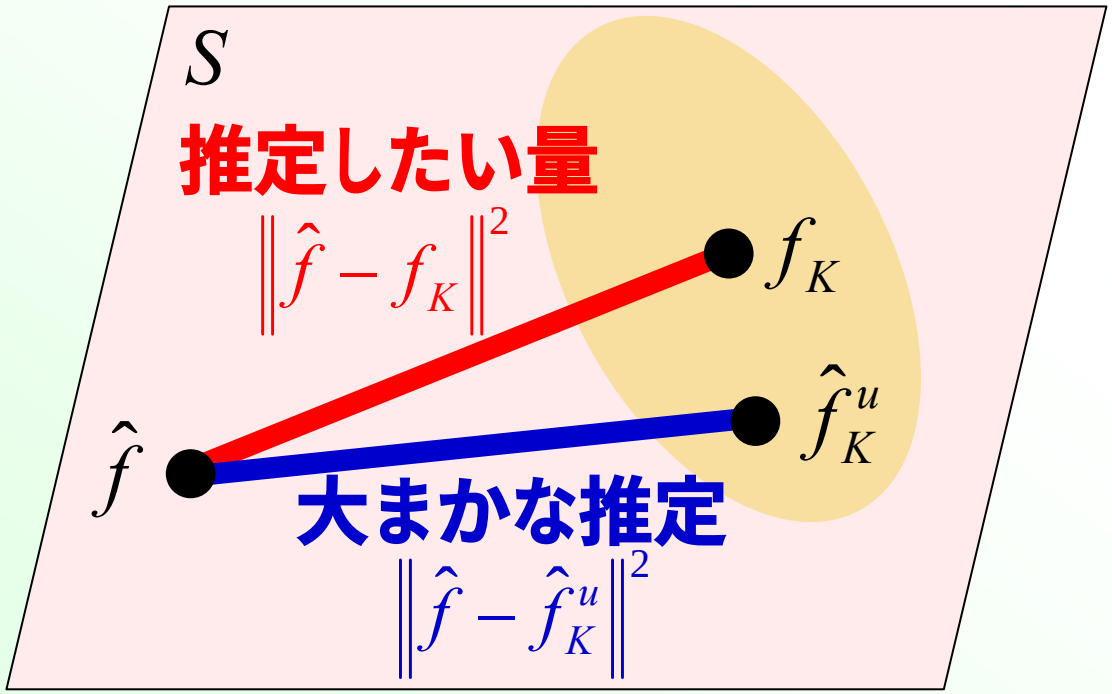
$S:\{K(\mathbf{x},\mathbf{x}_i)\}_{i=1}^{l}$ で張られる部分空間

$f_K:f$ の S への射影

SICのアイデア (2)

正射影 f_K は未知なので、
かわりに不偏推定量 $\hat{f}_K^u$ を利用する

$E\hat{f}_K^u = f_K$　　E : **雑音に関する平均**



$$\hat{f}_K^u(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i^u K(\mathbf{x}, \mathbf{x}_i)$$

$$\boldsymbol{\alpha}^u = \mathbf{K}^{-1}\mathbf{y}$$

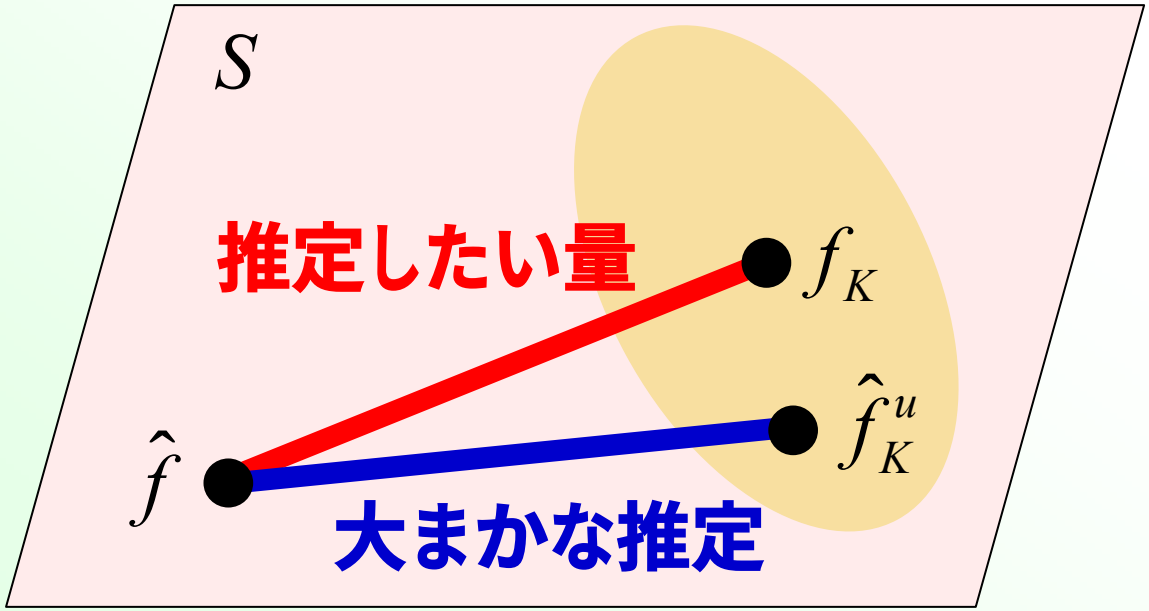
$$\boldsymbol{\alpha}^u = (\alpha_1^u, \alpha_2^u, \ldots, \alpha_n^u)^T$$

$$\mathbf{K}_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$$

$$\mathbf{y} = (y_1, y_2, \ldots, y_n)^T$$

SICのアイデア (3)

$$\underbrace{\left\|\hat{f}-f_K\right\|^2}_{\text{推定したい量}}=\underbrace{\left\|\hat{f}-\hat{f}_K^u\right\|^2}_{\text{大まかな推定}}+\underbrace{2\sum_{i=1}^{l}\varepsilon_i\alpha_i}_{2E\sum_{i=1}^{l}\varepsilon_i\alpha_i\text{で置き換え}}-\underbrace{\left\|\hat{f}_K^u\right\|^2}_{\text{計算できる}}+\underbrace{\left\|f_K\right\|^2}_{\text{定数(無視)}}$$



$f_K : f$ の S への射影

$\hat{f}_K^u : f_K$ の不偏推定量

$(E\hat{f}_K^u = f_K)$

E : 雑音に関する平均

SIC

$$\mathrm{SIC} = \left\| \hat{f} \right\|^2 - 2\left\langle \hat{f}, \hat{f}_S^u \right\rangle + 2E\sum_{i=1}^{l} \varepsilon_i \alpha_i$$

$$E\,\mathrm{SIC} = E\left\| \hat{f} - f \right\|^2 - \text{定数}$$

E : 雑音に関する平均

- 線形な学習法の場合 (従来のSIC)

  ➡ 解析的に計算

$$E\sum_{i=1}^{l} \varepsilon_i \alpha_i = \sigma^2 \sum_{i=1}^{l} c_{i,i}$$

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

$$\alpha_i = \sum_{j=1}^{l} c_{i,j} y_j$$

- 非線形な学習法の場合 (本研究)

  ➡ ブートストラップ法を用いて数値計算

ブートストラップ法

Efron (1979)

現実世界

$$\{\varepsilon_i\}_{i=1}^{l} \overset{i.i.d.}{\sim} \underbrace{p(\varepsilon)}_{\text{未知}}$$

何度も標本化できれば
精度よく推定できる

$$\underset{\varepsilon}{E}\,\theta\left(\{\varepsilon_i\}_{i=1}^{l}\right)$$

ブートストラップの世界

$$\{\varepsilon_i^b\}_{i=1}^{l} \overset{i.i.d.}{\sim} \underbrace{\hat{p}(\varepsilon)}_{\text{経験分布（既知）}}$$

何度でも標本化できるので
精度よく推定できる

$$\underset{\varepsilon^b}{E}\,\theta\left(\{\varepsilon_i^b\}_{i=1}^{l}\right)$$

“良い”
推定

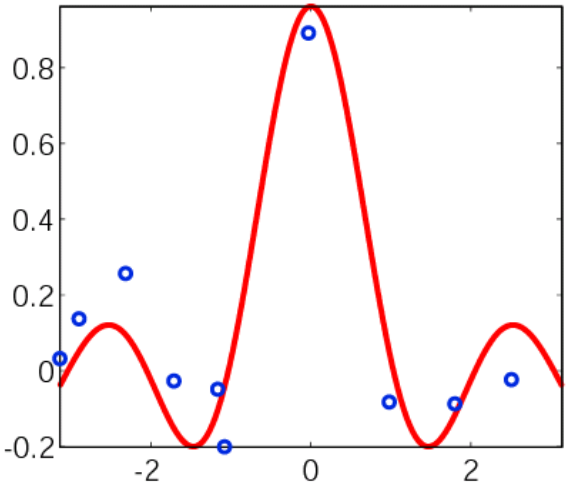
Bootstrap Approximation SIC

1. **訓練データ** $\{\mathbf{x}_i, y_i\}_{i=1}^{l}$ **を用いて**SVM**を学習し、**
   $\hat{f}(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i K(\mathbf{x}, \mathbf{x}_i)$ **を求める**
2. **雑音の推定値を求める**: $\hat{\varepsilon}_i = y_i - \hat{f}(\mathbf{x}_i)$
3. $\{\hat{\varepsilon}_i\}_{i=1}^{l}$ **から復元抽出し、**BS**複製** $\{\varepsilon_i^b\}_{i=1}^{l}$ **を生成する**
4. BS**訓練データ** $\left\{\mathbf{x}_i, \hat{y}_i \middle| \hat{y}_i = \hat{f}(\mathbf{x}_i) + \varepsilon_i^b\right\}_{i=1}^{l}$ **を用いて**
   SVM**を学習し、ＢＳ推定** $\{\hat{\alpha}_i^b\}_{i=1}^{l}$ **を求める**
5. 3.-4.**を繰り返す**

$$\mathrm{BASIC} = \left\|\hat{f}\right\|^2 - 2\left\langle \hat{f}, \hat{f}_S^u \right\rangle + 2 E_{\varepsilon^b} \sum_{i=1}^{l} \hat{\varepsilon}_i^b \hat{\alpha}_i^b$$

計算機実験

- 学習対象の関数 $f(x)$



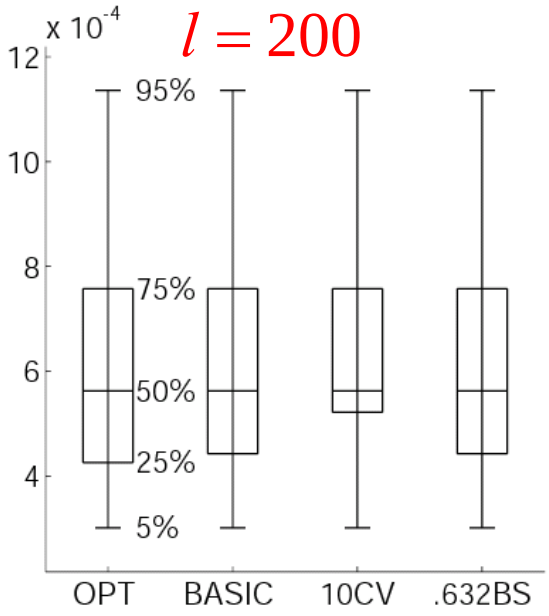
- $x_i : [-\pi, \pi]$ からランダムに発生
- $y_i = f(x_i) + \varepsilon_i$: $\varepsilon_i \overset{i.i.d.}{\sim} N(0, 0.01)$
- ガウシアンカーネル : $K(x, x') = \exp\left(-(x - x')^2 / 2\right)$
- 正則化パラメータ $C : \{10^{-2}, 10^{-1.5}, 10^{-1}, \ldots, 10^{0.5}\}$

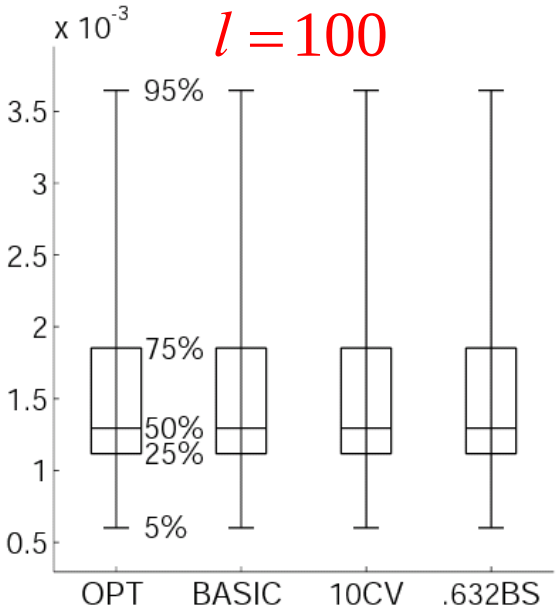
「BASIC」, 「10-fold CV」, 「.632」で C を選択し、
ランダムな1000点でのテストエラーを比較

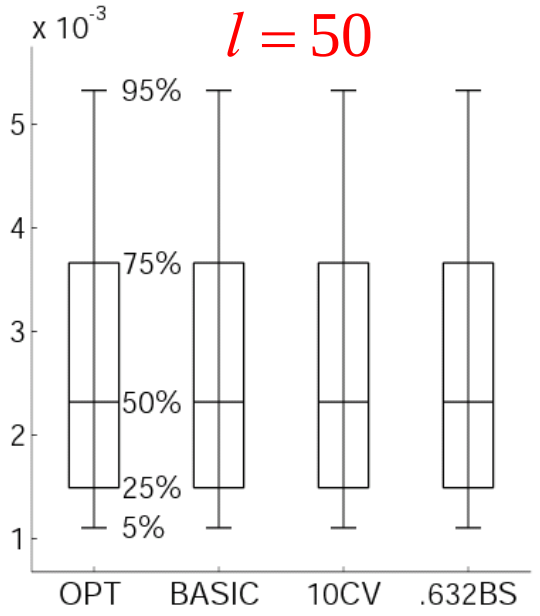
l : 訓練データ数
OPT : 最適

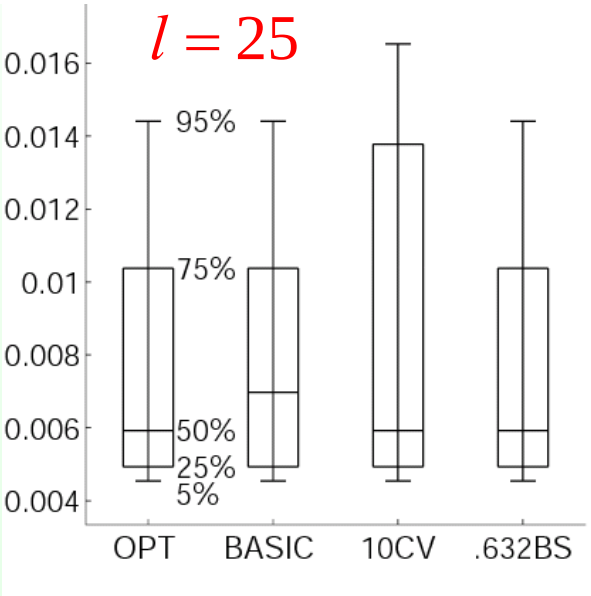
実験結果

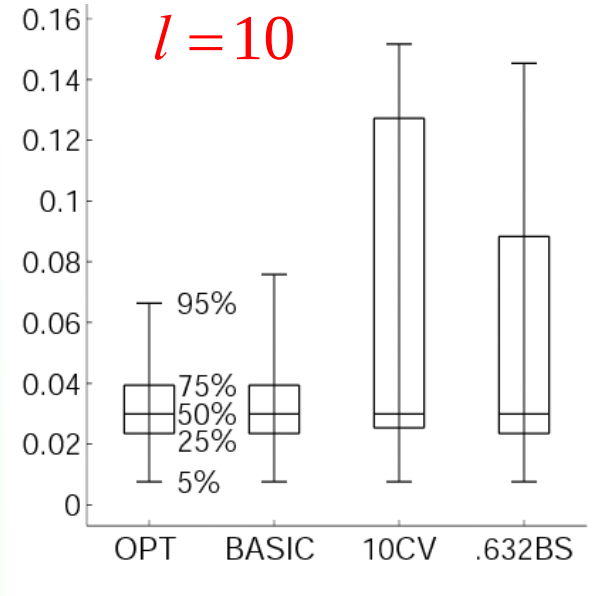
試行数 : 100











結論と今後の展望

- 関数解析的な立場からサポートベクター回帰の正則化パラメータを適切に決定する方法を提案した