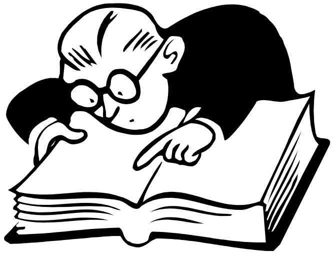
機械学習(1)：教師付き学習

杉山将・本多淳也
sugi@k.u-tokyo.ac.jp, jhonda@k.u-tokyo.ac.jp
http://www.ms.k.u-tokyo.ac.jp

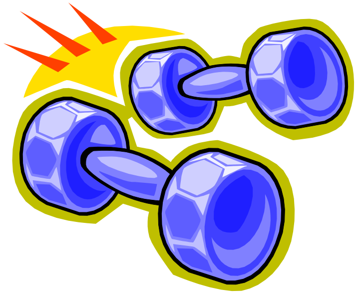
機械学習の分類

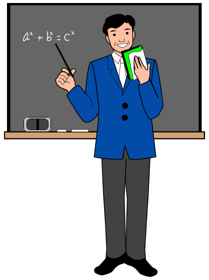
- 本講義では以下のトピックを扱う
  - 教師付き学習 (supervised learning):
    手持ちの正解付きデータを使って
    新たなデータに対する正解を予測する

  - 教師なし学習 (unsupervised learning):
    正解の与えられていないデータのみから
    意味のある情報を取り出す



  - 強化学習 (reinforcement learning):
    ロボット(等)に試行錯誤させることで
    よい行動を自動的に探索・学習させる

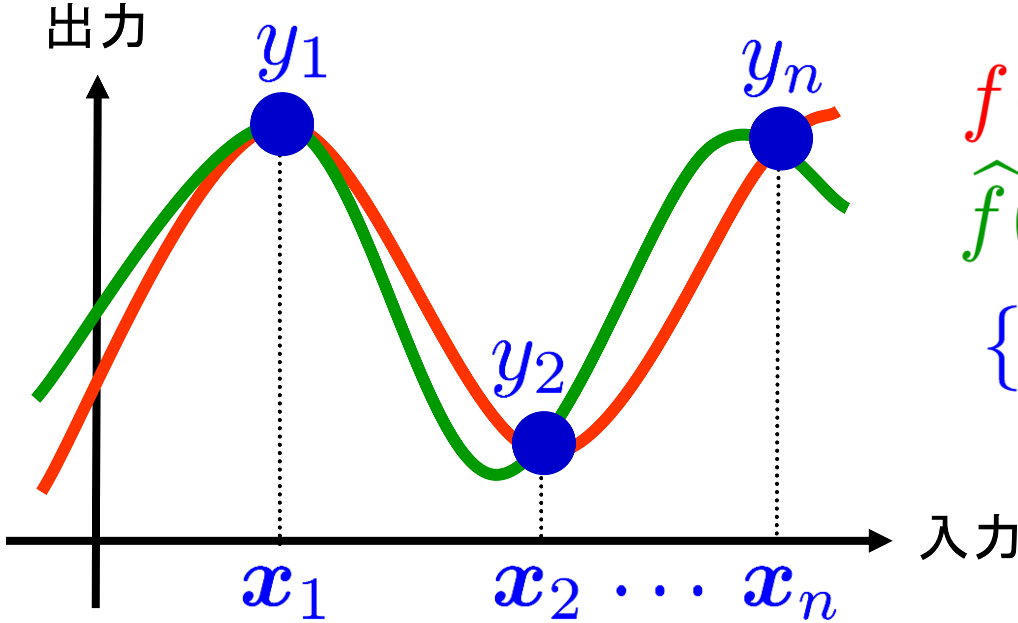




講義の流れ

1. 回帰
   - A) モデル
   - B) 最小二乗回帰
   - C) 正則化最小二乗回帰
   - D) 交差確認法
2. 分類

回帰 ＝ 関数近似



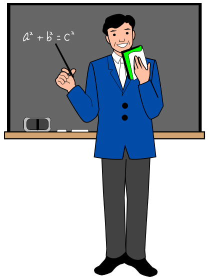
$f(\boldsymbol{x})$:学習したい真の関数

$\widehat{f}(\boldsymbol{x})$:学習結果の関数

$\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$:訓練標本

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

訓練標本から真の関数にできるだけ近い関数を求める



講義の流れ

1. 回帰
    - A) モデル
    - B) 最小二乗回帰
    - C) 正則化最小二乗回帰
    - D) 交差確認法
2. 分類

線形／非線形モデル

- モデル:学習結果の関数を探す候補集合
  - パラメータ $\boldsymbol{\theta}$ の値を指定すると関数が一つ決まる

$$\{f_{\boldsymbol{\theta}}(\boldsymbol{x}) \mid \boldsymbol{\theta} = (\theta_1, \ldots, \theta_b)^\top\}$$

- 線形モデル: $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ が $\boldsymbol{\theta}$ に関して線形
- 非線形モデル:それ以外(ニューラルネットなど)

線形モデル

- $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ が $\boldsymbol{\theta}$ に関して線形

- 一般形：

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{j=1}^{b} \theta_j \phi_j(\boldsymbol{x})$$

$$\boldsymbol{x} \in \mathbb{R}^d$$

- $\{\phi_j(\boldsymbol{x})\}_{j=1}^{b}$：線形独立な（既知の）基底関数
  - 例（ $d=1$ 次元の場合）：多項式基底

$$1, x, x^2, \ldots, x^{b-1}$$

- より狭義には入力 $\boldsymbol{x}$ に対して線形，すなわち $\phi_j(\boldsymbol{x}) = x_j$ の場合を指すこともある

カーネルモデル

- 線形モデルの一種，基底関数が入力に依存：

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{j=1}^{n} \theta_j K(\boldsymbol{x}, \boldsymbol{x}_j)$$

$\boldsymbol{\theta} = (\theta_1, \ldots, \theta_n)^\top$：パラメータ

$K(\boldsymbol{x}, \boldsymbol{c})$ ：カーネル関数

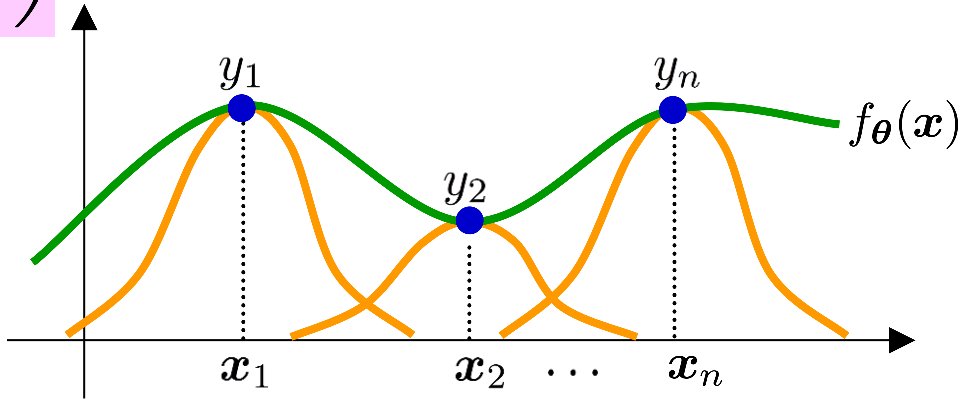
  - $\phi_j(\cdot) = K(\,\cdot\,, \boldsymbol{x}_j)$ に対応

- ガウスカーネル：

$$K(\boldsymbol{x}, \boldsymbol{c}) = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{c}\|^2}{2h^2}\right)$$

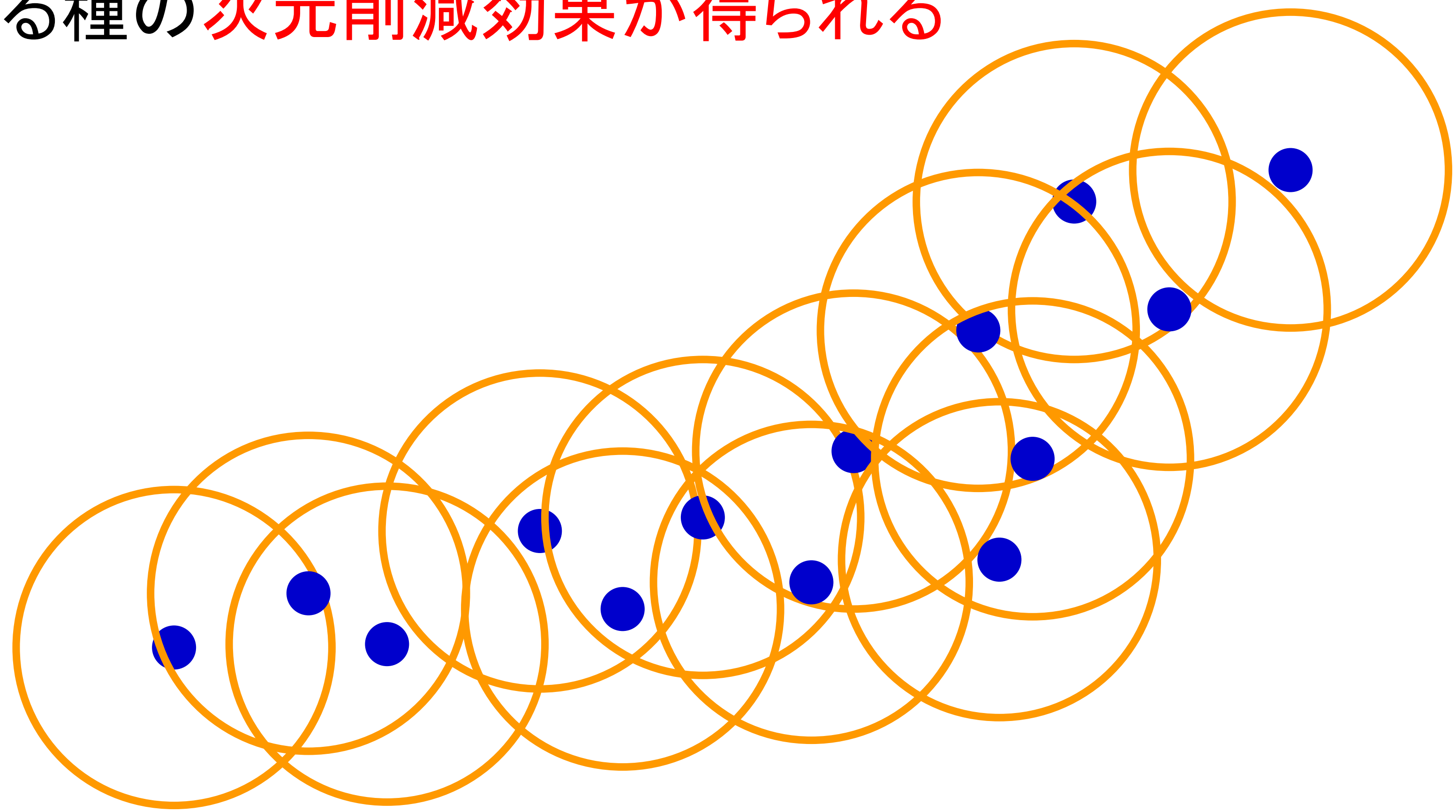
$h(>0)$：バンド幅

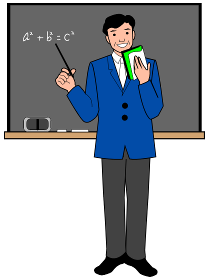
  - ガウス関数を各訓練入力標本の場所に配置



ガウスカーネルモデル

- 訓練標本が入力空間上に偏って分布している時，ガウスカーネルモデルは訓練入力標本が存在しない領域を無視する関数が学習される
  - ある種の次元削減効果が得られる





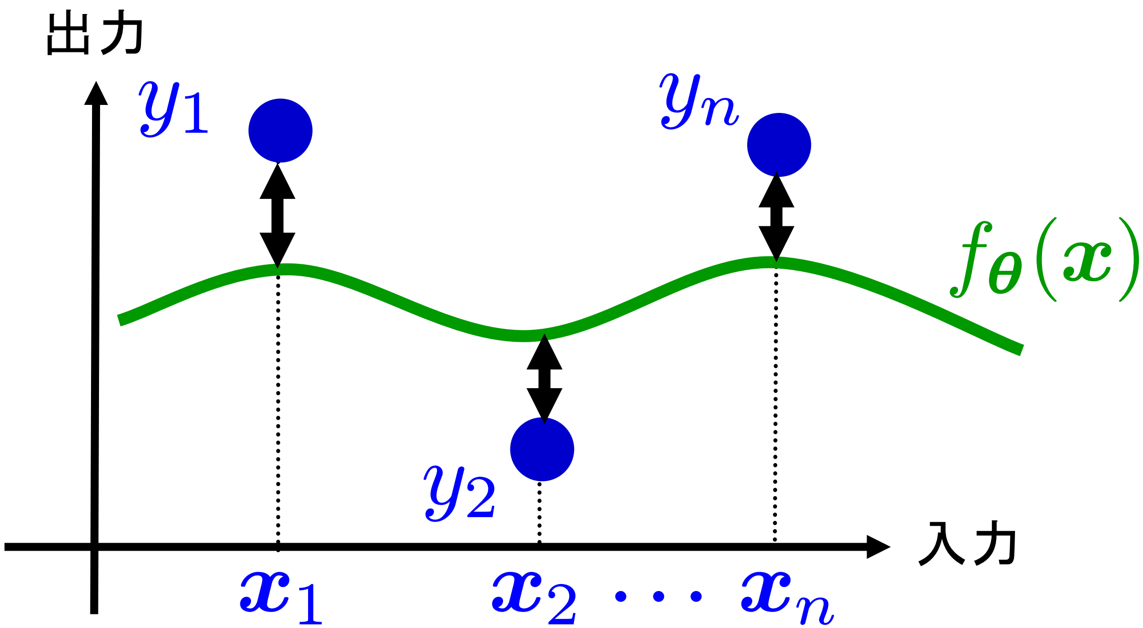
講義の流れ

1. 回帰
   - A) モデル
   - B) 最小二乗回帰
   - C) 正則化最小二乗回帰
   - D) 交差確認法
2. 分類

最小二乗回帰

- 規準：訓練出力との二乗誤差を最小にする

$$\min_{\boldsymbol{\theta}} \left[\frac{1}{2} \sum_{i=1}^{n} \left(y_i - f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) \right)^2 \right]$$



解の求め方

- 行列・ベクトル表現: $\|\boldsymbol{a}\|^2 = \sum_{i=1}^{n} a_i^2 = \boldsymbol{a}^\top \boldsymbol{a}$ に対して

$$\frac{1}{2}\sum_{i=1}^{n}\Big(y_i - f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\Big)^2 = \frac{1}{2}\sum_{i=1}^{n}\left(y_i - \sum_{j=1}^{n}\theta_j K(x_i, x_j)\right)^2$$

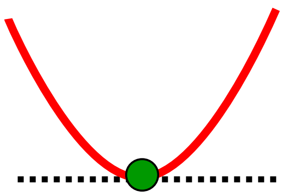
$$= \frac{1}{2}\|\boldsymbol{K}\boldsymbol{\theta} - \boldsymbol{y}\|^2 = \frac{1}{2}(\boldsymbol{K}\boldsymbol{\theta} - \boldsymbol{y})^\top(\boldsymbol{K}\boldsymbol{\theta} - \boldsymbol{y})$$

$$\boldsymbol{\theta} = (\theta_1, \ldots, \theta_n)^\top \quad \boldsymbol{y} = (y_1, \ldots, y_n)^\top$$

$$\boldsymbol{K} = \begin{pmatrix} K(\boldsymbol{x}_1, \boldsymbol{x}_1) & \cdots & K(\boldsymbol{x}_1, \boldsymbol{x}_n) \\ \vdots & \ddots & \vdots \\ K(\boldsymbol{x}_n, \boldsymbol{x}_1) & \cdots & K(\boldsymbol{x}_n, \boldsymbol{x}_n) \end{pmatrix}$$:カーネル行列

解の求め方(続き)

■ 偏微分がゼロの点を求める:



$$\nabla_{\boldsymbol{\theta}} \left(\frac{(\boldsymbol{K\theta} - \boldsymbol{y})^\top (\boldsymbol{K\theta} - \boldsymbol{y})}{2} \right) = \boldsymbol{K}(\boldsymbol{K\theta} - \boldsymbol{y}) = \boldsymbol{0}$$

- 微分公式: $\frac{\partial}{\partial \boldsymbol{\theta}} \boldsymbol{\theta}^\top \boldsymbol{A\theta} = 2\boldsymbol{A\theta}$ $\frac{\partial}{\partial \boldsymbol{\theta}} \boldsymbol{b}^\top \boldsymbol{\theta} = \boldsymbol{b}$

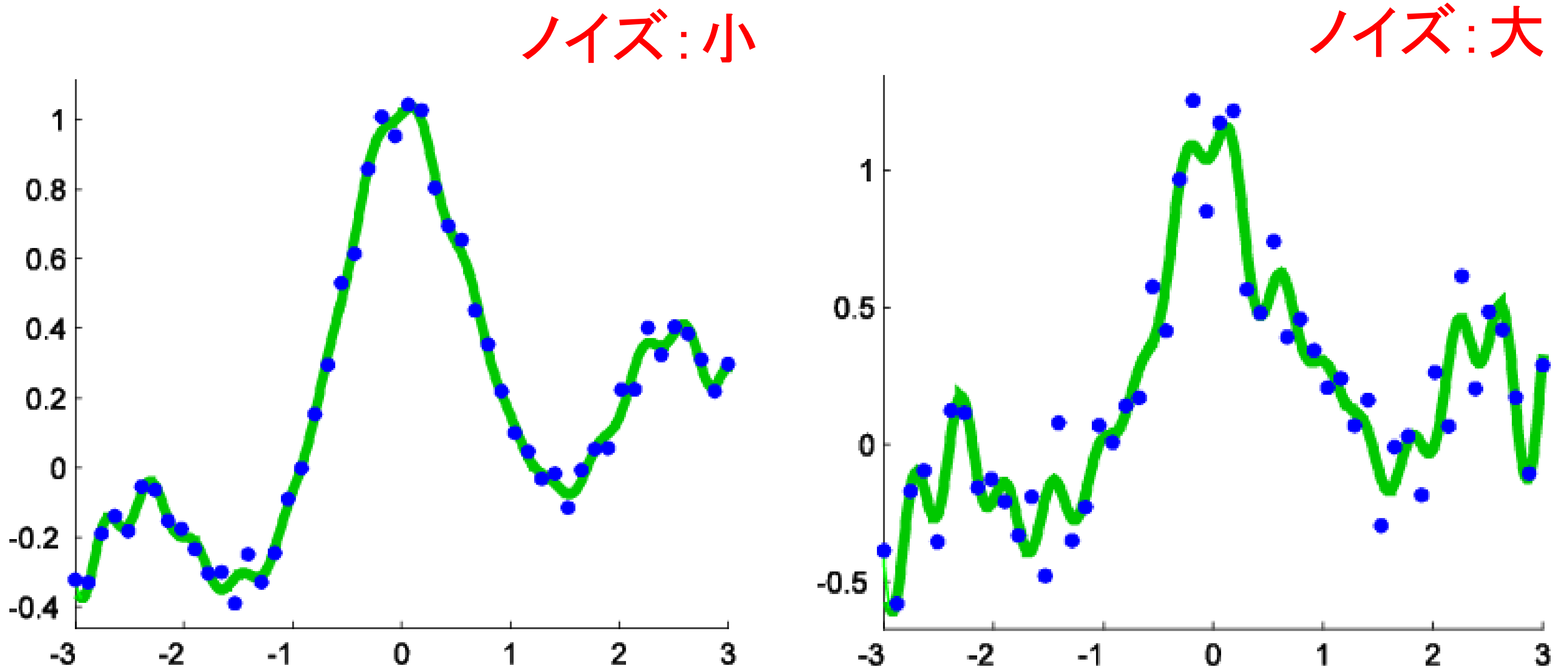
$$\Longrightarrow \quad \boldsymbol{K}^2 \boldsymbol{\theta} = \boldsymbol{K y}$$

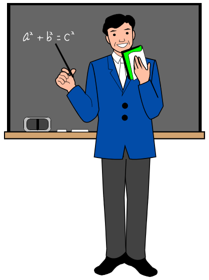
■ 解を解析的に求められる!

$$\widehat{\boldsymbol{\theta}} = \boldsymbol{K}^{-1} \boldsymbol{y}$$

最小二乗回帰の問題点

- 過適合(overfitting):
  ノイズを含む訓練標本に過剰に適合する





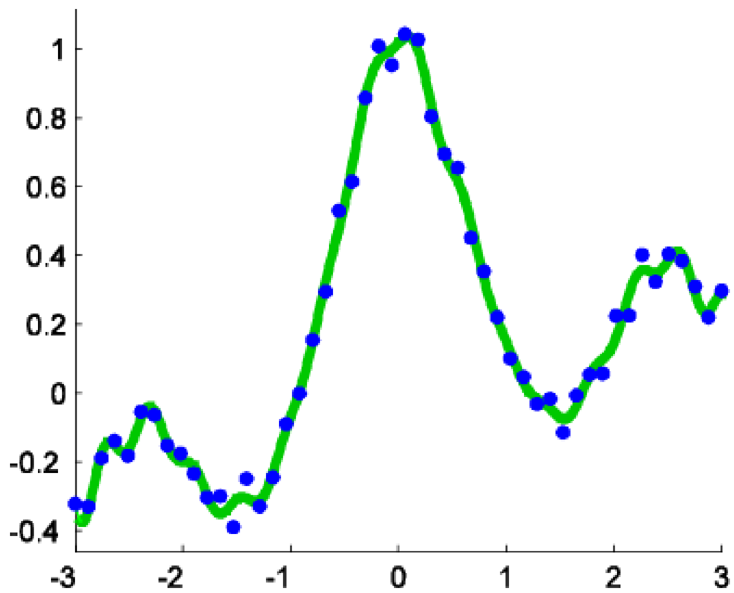
講義の流れ

1. 回帰
   - A) モデル
   - B) 最小二乗回帰
   - C) 正則化最小二乗回帰
   - D) 交差確認法
2. 分類

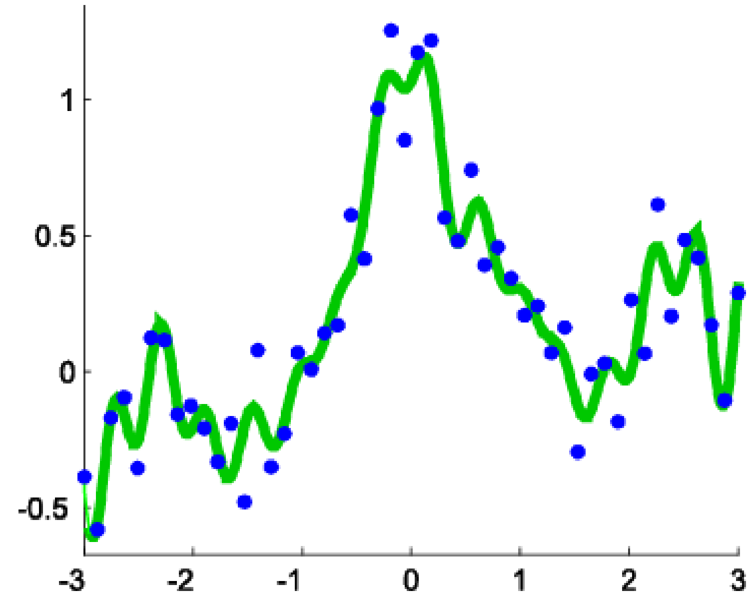
過適合の対策

- 過適合した回帰曲線は一般にギザギザした形である場合が多い

ノイズ：小



ノイズ：大



- 回帰係数 θ の各要素の絶対値が大きくなることを許すとギザギザした回帰曲線を作れてしまう

正則化最小二乗回帰

$$\underbrace{\frac{1}{2}\sum_{i=1}^{n}\Big(y_i - f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\Big)^2}_{\text{訓練出力に対する適合の良さ}} + \underbrace{\frac{\lambda}{2}\sum_{j=1}^{n}\theta_j^2}_{\text{パラメータの値が大きくなり過ぎることに対する罰則（正則化）}} \qquad \lambda \geq 0$$

- 過適合を避けるため,
  - 訓練出力に対する適合のよさ
  - パラメータの値の大きさ

  をバランスよく小さくする

数学演習

- 正則化最小二乗回帰の解を求めよ
  - ヒント：行列・ベクトル表現：

$$\frac{1}{2}\sum_{i=1}^{n}\Big(y_i - f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\Big)^2 + \frac{\lambda}{2}\sum_{j=1}^{n}\theta_j^2$$

$$= \frac{1}{2}\|\boldsymbol{K\theta} - \boldsymbol{y}\|^2 + \frac{\lambda}{2}\|\boldsymbol{\theta}\|^2$$

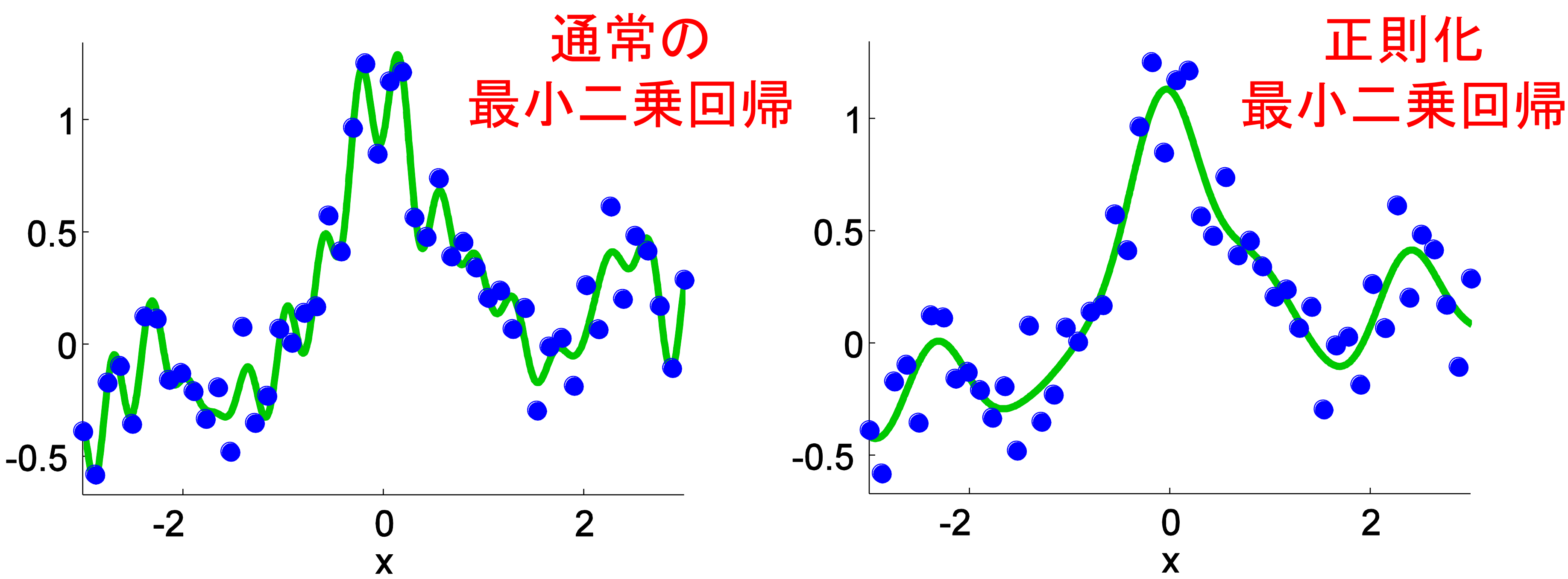
$$\boldsymbol{\theta} = (\theta_1, \ldots, \theta_n)^\top \qquad \boldsymbol{y} = (y_1, \ldots, y_n)^\top$$

$$\boldsymbol{K} = \begin{pmatrix} K(\boldsymbol{x}_1, \boldsymbol{x}_1) & \cdots & K(\boldsymbol{x}_1, \boldsymbol{x}_n) \\ \vdots & \ddots & \vdots \\ K(\boldsymbol{x}_n, \boldsymbol{x}_1) & \cdots & K(\boldsymbol{x}_n, \boldsymbol{x}_n) \end{pmatrix}$$:カーネル行列

実行例

$$\min_{\boldsymbol{\theta}} \left[\frac{1}{2} \sum_{i=1}^{n} \Big(y_i - f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) \Big)^2 + \frac{\lambda}{2} \sum_{j=1}^{n} \theta_j^2 \right]$$

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{j=1}^{n} \theta_j \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}_j\|^2}{2h^2} \right)$$



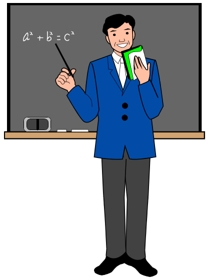
- 過適合が回避できている

Octaveのプログラム例


```
clear all; rand('state',0); randn('state',0);
n=50; N=1000;
x=linspace(-3,3,n)'; X=linspace(-3,3,N)';
pix=pi*x; y=sin(pix)./(pix)+0.1*x+0.2*randn(n,1);

x2=x.^2; hh=2*0.3^2; l=0.1;
k=exp(-(repmat(x2,1,n)+repmat(x2',n,1)-2*x*x')/hh);
t=(k^2+l*eye(n))\(k*y);

X2=X.^2;
K=exp(-(repmat(X2,1,n)+repmat(x2',N,1)-2*X*x')/hh);
F=K*t;
figure(1); clf; hold on; axis([-2.8 2.8 -1 1.5]);
plot(X,F,'g-'); plot(x,y,'bo');
```

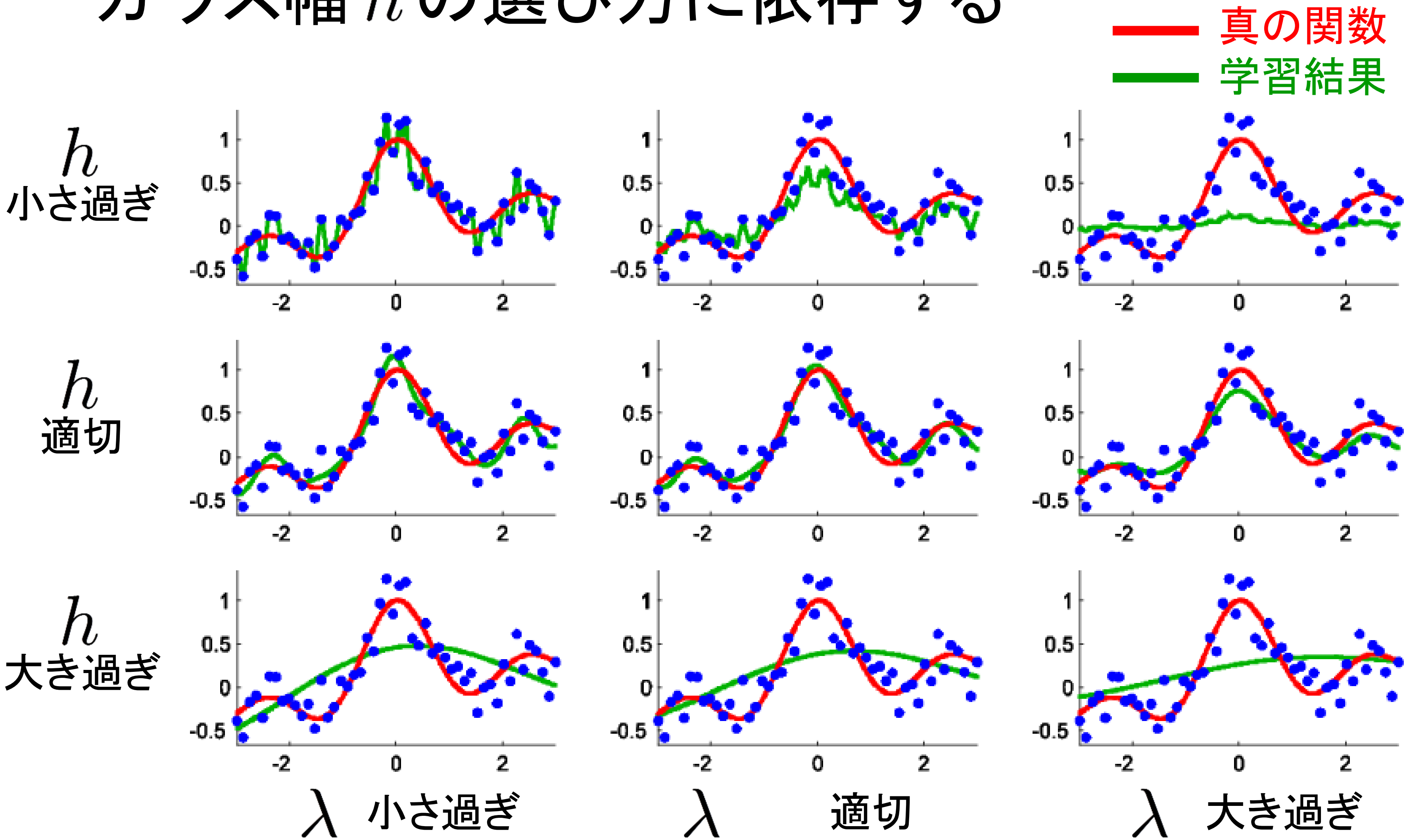


講義の流れ

1. 回帰
   - A) モデル
   - B) 最小二乗回帰
   - C) 正則化最小二乗回帰
   - D) 交差確認法
2. 分類

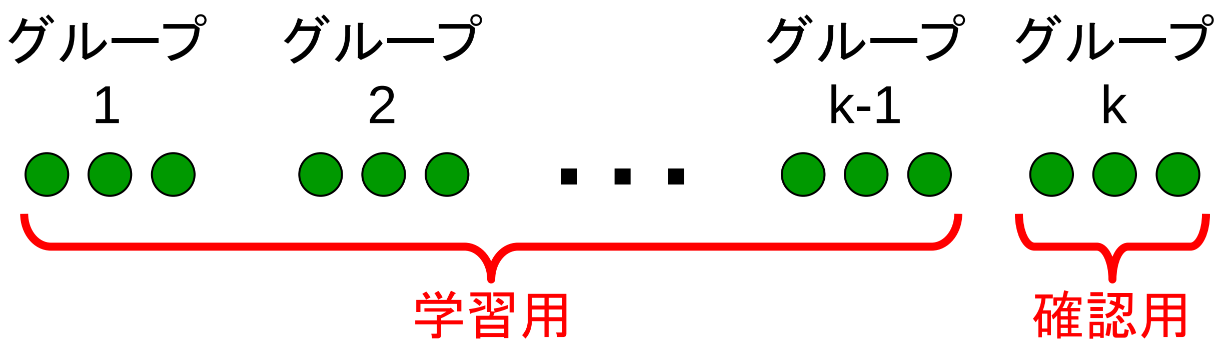
正則化回帰のモデル選択

- 正則化回帰の結果は，正則化パラメータλと
ガウス幅hの選び方に依存する



交差確認法

- 訓練標本 $\mathcal{Z} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ をk分割する: $\{\mathcal{Z}_i\}_{i=1}^k$
- $\mathcal{Z}_i$ 以外を使って関数を学習する
- 残った $\mathcal{Z}_i$ を使ってテスト誤差を確認する
- これを全ての組み合わせに対して繰り返し，平均を出力する

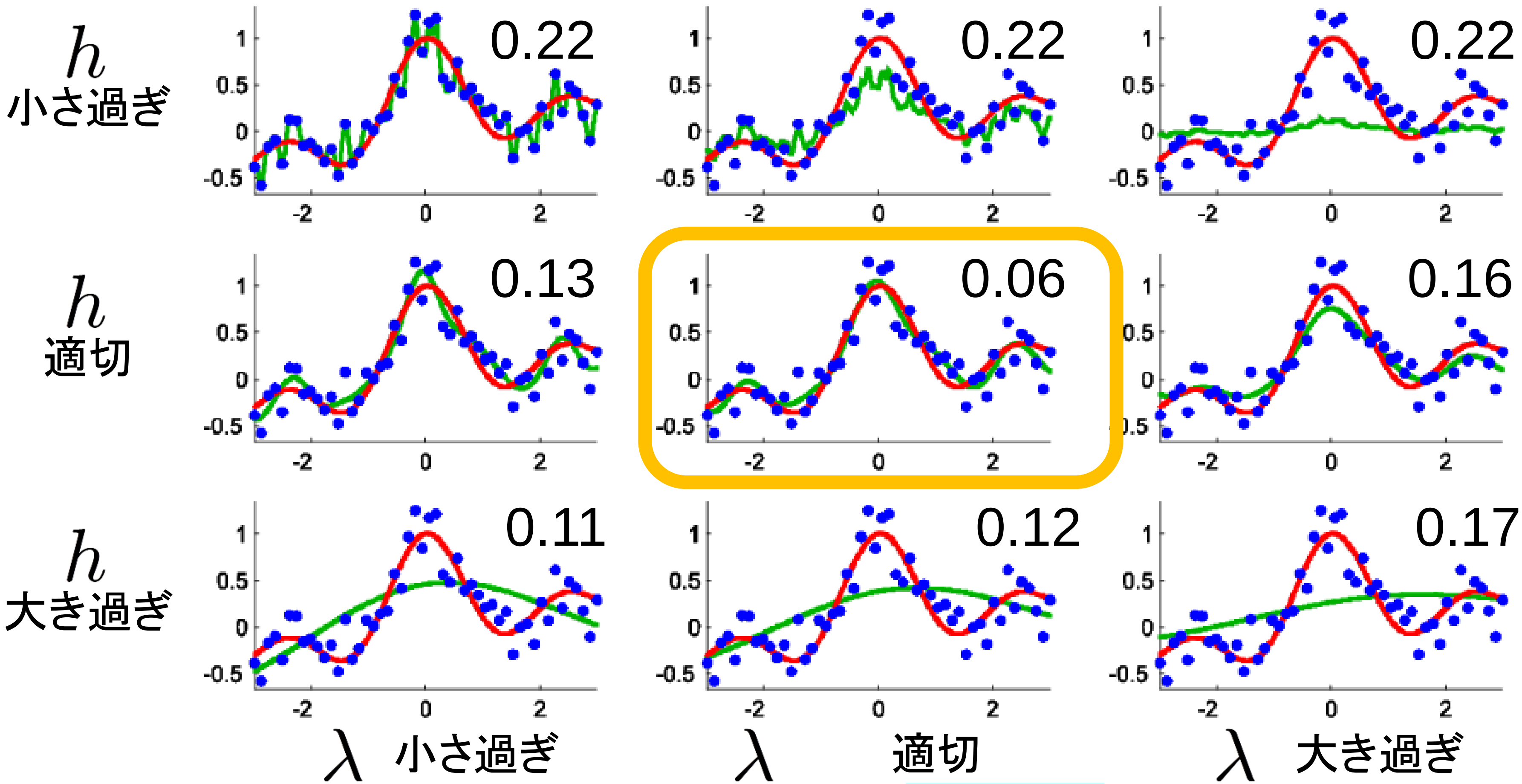


$$\widehat{f}(\boldsymbol{x})$$

$$\left(\widehat{f}(\boldsymbol{x}') - y'\right)^2$$

交差確認法の実行例

- ガウスカーネルモデル: $f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{j=1}^{n} \theta_j \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}_j\|^2}{2h^2}\right)$



- 妥当な結果が得られている

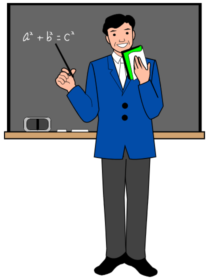
実装は宿題

真の関数
学習結果

回帰のまとめ

- 最小二乗回帰は雑音に過適合しやすい
  - パラメータの二乗和を正則化項として加える
- 解は解析的に求められる
- モデル選択が重要
  - 交差確認法が実用的
- 損失と正則化を変えると色々な学習法が作れる
  - ロバスト回帰（損失をなだらかな関数に）
  - スパース回帰（ゼロが出やすい正則化項を使う）

$$\min_{\boldsymbol{\theta}} \left[\sum_{i=1}^{n} \mathrm{loss}\Big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), y_i \Big) + \lambda \mathrm{Reg}(\boldsymbol{\theta}) \right]$$

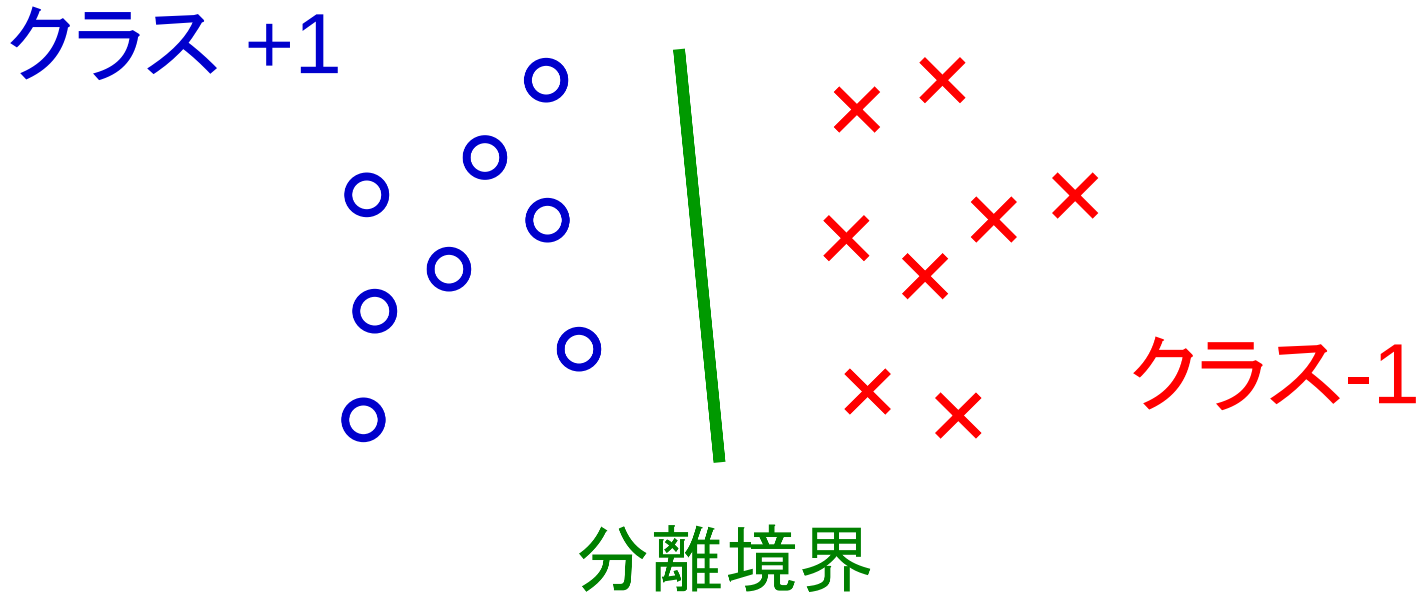


講義の流れ

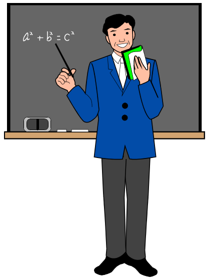
1. 回帰
2. 分類
   - A) 超平面分類器
   - B) サポートベクトルマシン
   - C) 非線形化
   - D) 最適化

2クラスの分類問題

- ラベル付き訓練データ: $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$
  - 入力 $\boldsymbol{x}$ は d 次元の実ベクトル $\boldsymbol{x} \in \mathbb{R}^d$
  - 出力 y は2値のクラスラベル $y \in \{+1, -1\}$



- クラス間の分離境界を求めたい



講義の流れ

1. 回帰
2. 分類
   - A) 超平面分類器
   - B) サポートベクトルマシン
   - C) カーネルトリック
   - D) 最適化

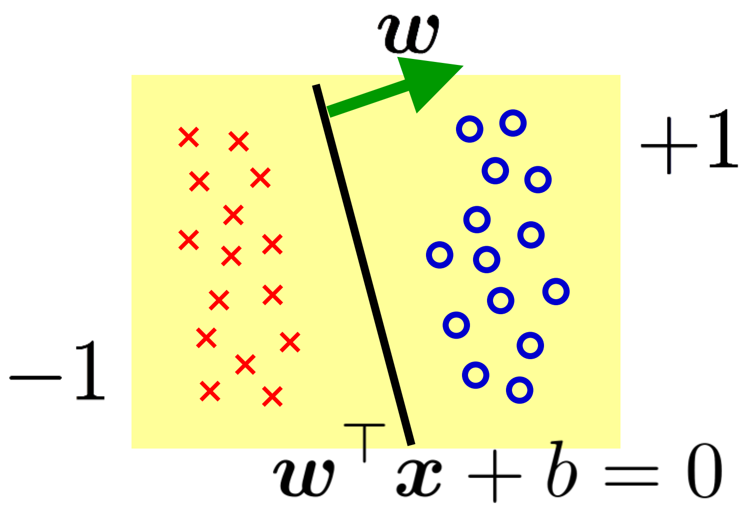
超平面分類器

- 標本空間を超平面で分離する.

$$f_{\boldsymbol{w},b}(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + b$$

$$\boldsymbol{w} = (w^{(1)}, \ldots, w^{(d)})^\top$$

$$\boldsymbol{x} = (x^{(1)}, \ldots, x^{(d)})^\top$$



find $\boldsymbol{w}, b$

such that $\quad y_i f_{\boldsymbol{w},b}(\boldsymbol{x}_i) > 0 \quad$ for $i = 1, \ldots, n.$

超平面分類器

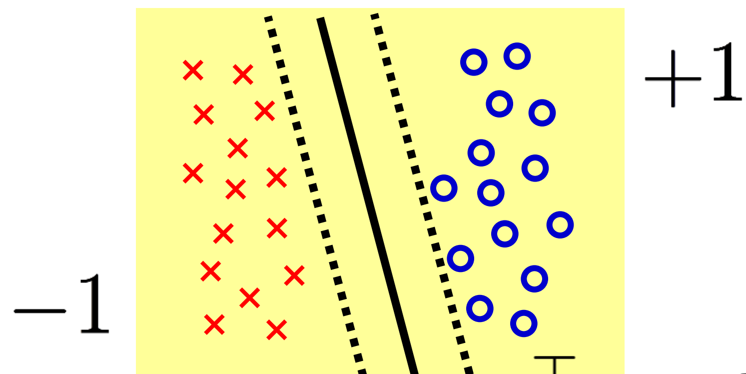
- これは適当なスケーリングのもと以下と等価.

$$f_{\boldsymbol{w},b}(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + b$$

$$\boldsymbol{w} = \left(w^{(1)}, \ldots, w^{(d)}\right)^\top$$

$$\boldsymbol{x} = \left(x^{(1)}, \ldots, x^{(d)}\right)^\top$$

$$\boldsymbol{w}^\top \boldsymbol{x} + b = -1$$



$$\boldsymbol{w}^\top \boldsymbol{x} + b = 1$$

$$\text{find } \boldsymbol{w}, b$$

$$\text{such that} \quad y_i f_{\boldsymbol{w},b}(\boldsymbol{x}_i) \geq 1 \quad \text{for } i = 1, \ldots, n.$$

詳細な導出

- もし $y_i f_{\boldsymbol{w},b}(\boldsymbol{x}_i) > 0 \quad \text{for } i = 1, \ldots, n$ となる $\boldsymbol{w}, b$ が存在するなら, n が有限なので

$$\epsilon := \min_{i \in \{1,\ldots,n\}} \{y_i f_{\boldsymbol{w},b}(\boldsymbol{x}_i)\} > 0$$

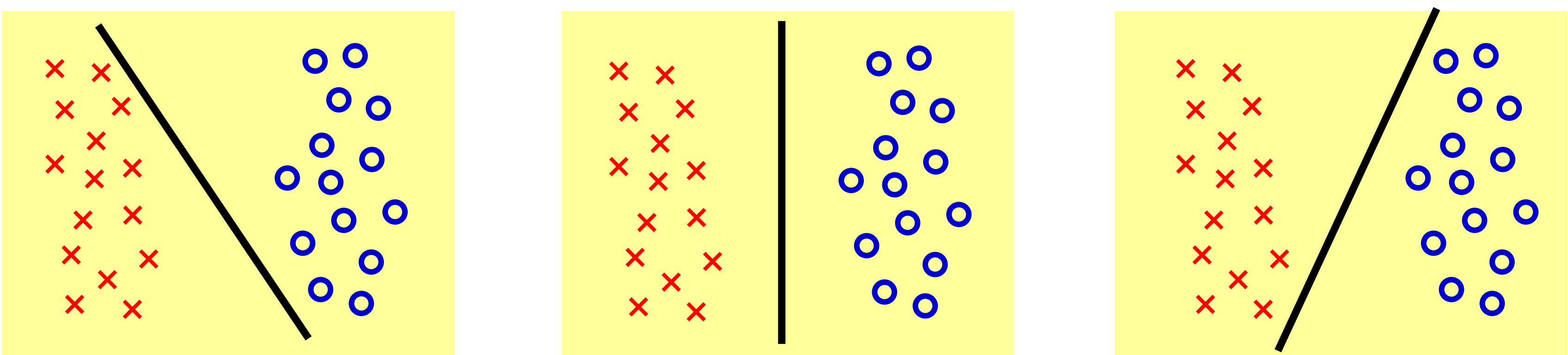
- $\boldsymbol{w}' = \boldsymbol{w}/\epsilon,\ b' = b/\epsilon$ とおくと

$$\begin{aligned} y_i f_{\boldsymbol{w}',b'}(\boldsymbol{x}_i) &= y_i((\boldsymbol{w}')^\top \boldsymbol{x}_i + b') \\ &= y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b)/\epsilon \\ &= y_i f_{\boldsymbol{w},b}(\boldsymbol{x}_i)/\epsilon \\ &\geq \epsilon/\epsilon = 1 \end{aligned}$$

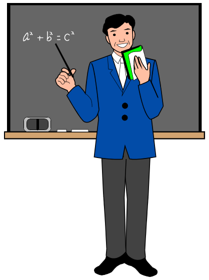
  - すなわち $y_i f_{\boldsymbol{w}',b'}(\boldsymbol{x}_i) \geq 1 \quad \text{for } i = 1, \ldots, n$ が成立

最適な分離は？

- 以下の3つの分け方で，将来与えられるテストデータに対する分類誤差（汎化誤差）を最小にするものはどれか？



- 3つとも，学習用データを全て正しく分離できるため，優劣をつけがたい．
- 真ん中の分離平面が直感的にはよさそう

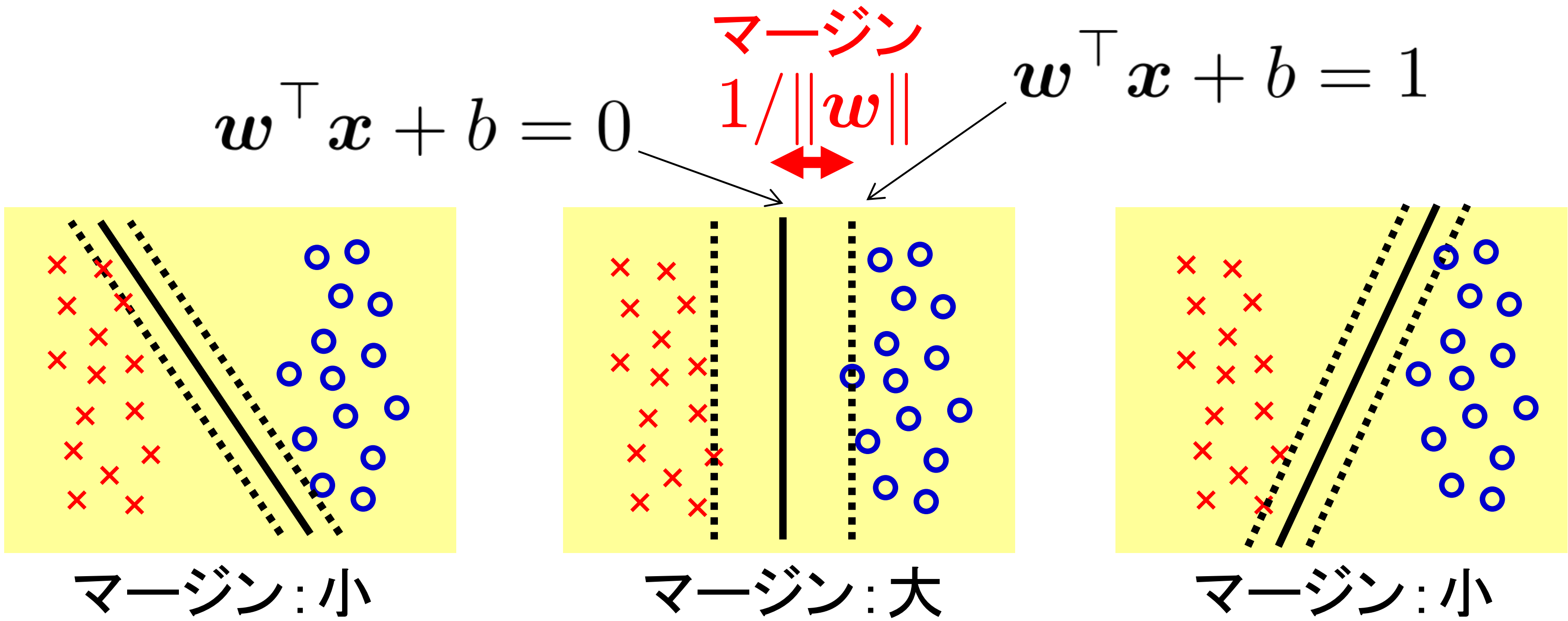


講義の流れ

1. 回帰
2. 分類
   A) 超平面分類器
   B) サポートベクトルマシン
   C) カーネルトリック
   D) 最適化

マージン

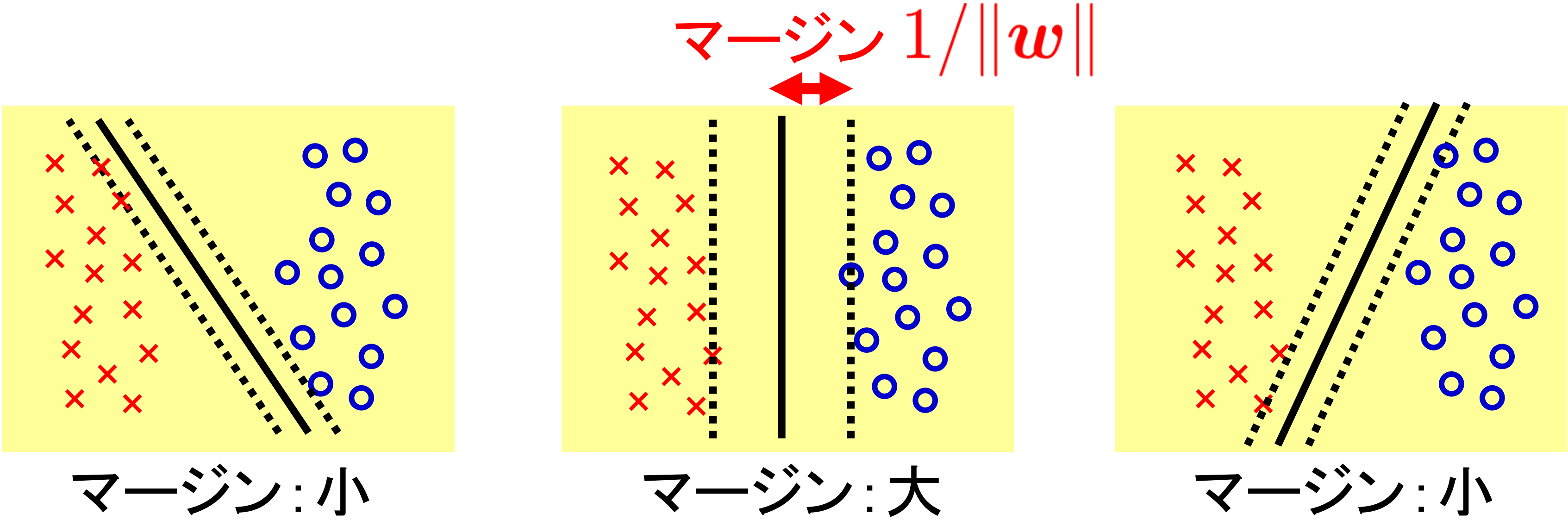
- マージン：二つのクラスの“隙間”の大きさ



- マージンが大きい分類器は訓練データのずれに対して頑健→汎化誤差が小さくなりやすい（より詳細にはVC次元の理論等を用いる）

最適超平面分類器

- マージンが最大になるように二つのクラスを超平面で分ける.



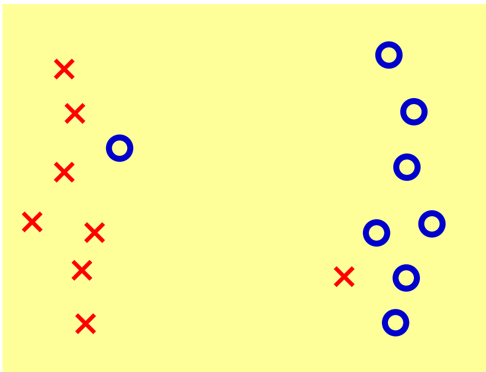
$$\min_{\boldsymbol{w},b} \|\boldsymbol{w}\|^2$$

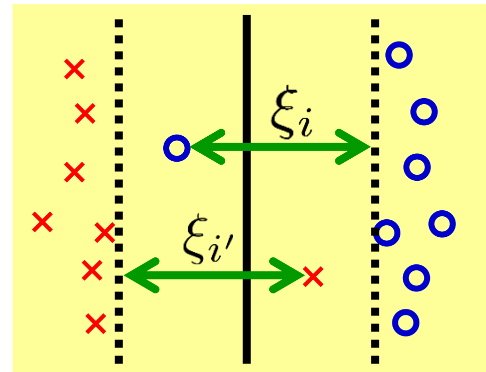
$$f_{\boldsymbol{w},b}(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + b$$

$$\text{subject to} \quad y_i f_{\boldsymbol{w},b}(\boldsymbol{x}_i) \geq 1 \quad \text{for } i = 1, \ldots, n.$$

ソフトマージン

- 標本が線形分離可能でないときはマージンが定義できないため，少しの**誤差** ξ_i を許す．





$$\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}\xi_i$$

$$f_{\boldsymbol{w},b}(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + b$$

$$\text{subject to} \quad y_i f_{\boldsymbol{w},b}(\boldsymbol{x}_i) \geq 1 - \xi_i$$

$$\xi_i \geq 0 \quad \text{for } i = 1, \ldots, n.$$

- この手法を**サポートベクトルマシン**とよぶ

制約なし最適化への変形

- サポートベクトルマシンでの計算:

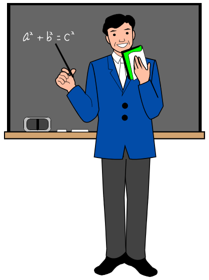
$$\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}\xi_i \qquad f_{\boldsymbol{w},b}(\boldsymbol{x}) = \boldsymbol{w}^\top\boldsymbol{x} + b$$

$$\text{subject to} \quad y_i f_{\boldsymbol{w},b}(\boldsymbol{x}_i) \geq 1 - \xi_i$$

$$\xi_i \geq 0 \quad \text{for } i = 1,\ldots,n.$$

- 最初の制約式は $\xi_i \geq 1 - y_i f_{\boldsymbol{w},b}(\boldsymbol{x}_i)$ と等価
- ξ_iが小さいほど目的関数が減るので,
  結局必要な計算は

$$\min_{\boldsymbol{w},b} \|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}\max\{0, 1 - y_i f_{\boldsymbol{w},b}(\boldsymbol{x})\}$$

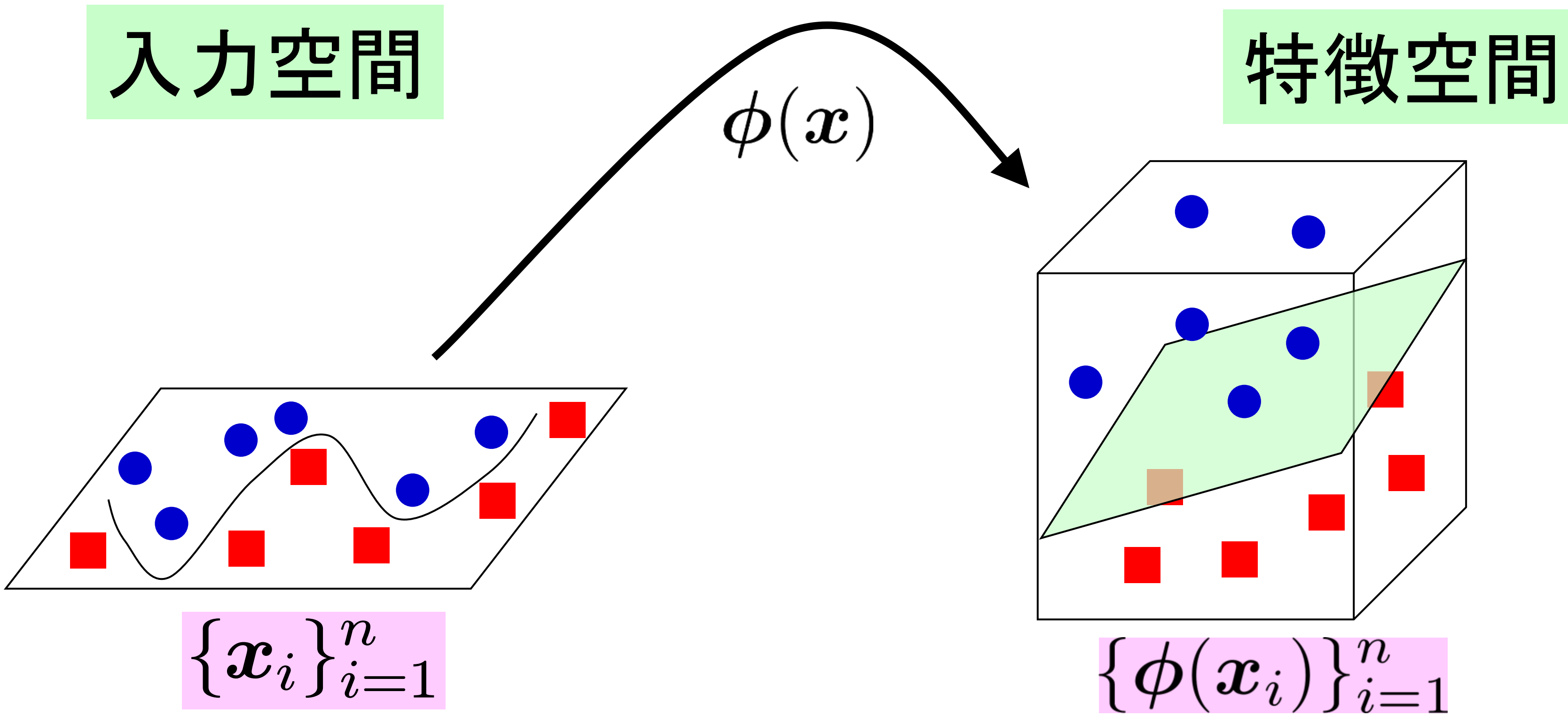


講義の流れ

1. 回帰
2. 分類
   - A) 超平面分類器
   - B) サポートベクトルマシン
   - C) カーネルトリック
   - D) 最適化

非線形への拡張

- 非線形関数 $\phi(\boldsymbol{x})$ で標本を特徴空間へ写像し，特徴空間内でマージン最大の超平面を求める.



非線形の基底を用いる場合

- サポートベクトルマシンでの計算:

$$\min_{\boldsymbol{w},b} \|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n} \max\{0, 1 - y_i f_{\boldsymbol{w},b}(\boldsymbol{x}_i)\}$$

$$f_{\boldsymbol{w},b}(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x}) + b$$

  - これをそのまま解いてもよいが，$\phi(x)$ の次元が大きい場合は（無限大の場合も）計算が大変/過適合が起きやすい

非線形の基底を用いる場合

- サポートベクトルマシンでの計算:

$$\min_{\boldsymbol{w},b} \|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n} \max\{0, 1 - y_i f_{\boldsymbol{w},b}(\boldsymbol{x}_i)\}$$

$$f_{\boldsymbol{w},b}(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x}) + b$$

- $\boldsymbol{w} = \sum_{j=1}^{n} \theta_j \boldsymbol{\phi}(\boldsymbol{x}_j)$ と限定すると, 以下の最適化に

$$\min_{\boldsymbol{\theta}} \left[C\sum_{i=1}^{n} \max\Big(0, 1 - f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) y_i\Big) + \boldsymbol{\theta}^\top \boldsymbol{K} \boldsymbol{\theta} \right]$$

$$\boldsymbol{\phi}(\boldsymbol{x}_i)^\top \boldsymbol{\phi}(\boldsymbol{x}_j) = K(\boldsymbol{x}_i, \boldsymbol{x}_j) \qquad f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{j=1}^{n} \theta_j K(\boldsymbol{x}, \boldsymbol{x}_j) + \theta_0$$

カーネルトリック

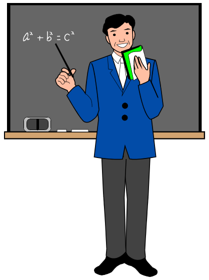
- 線形モデルにおける多くの回帰・分類手法では特徴量 $\phi(\boldsymbol{x}_i)$ を直接求めなくても，その内積 $\phi(\boldsymbol{x}_i)^\top \phi(\boldsymbol{x}_j)$ さえわかれば実装可能
- 特徴空間上の内積をカーネル関数で直接表現：

$$\phi(\boldsymbol{x}_i)^\top \phi(\boldsymbol{x}_j) = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

  - 例：ガウスカーネル

$$K(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\|\boldsymbol{x} - \boldsymbol{x}'\|^2 / (2h^2)\right) \geq 0$$

- 計算量が特徴空間の次元によらない
  - ガウスカーネルに対応する特徴空間は無限次元
  - ただし標本数 n に対する計算量は悪化



講義の流れ

1. 回帰
2. 分類
   - A) 超平面分類器
   - B) サポートベクトルマシン
   - C) カーネルトリック
   - D) 最適化

劣勾配法による実装

- カーネルトリックを用いた場合の最適化:

$$\min_{\boldsymbol{\theta}} \left[C \sum_{i=1}^{n} \max\left(0, 1 - f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) y_i\right) + \boldsymbol{\theta}^\top \boldsymbol{K} \boldsymbol{\theta} \right]$$

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{j=1}^{n} \theta_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) + \theta_0$$

（以降 θ_0 は便宜上無視する）

- 劣勾配法で最適化できる:

$$\boldsymbol{\theta} \longleftarrow \boldsymbol{\theta} - \varepsilon \left(C \sum_{i=1}^{n} \underbrace{\partial_{\boldsymbol{\theta}} \max\left(0, 1 - f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) y_i\right)}_{\text{劣微分}} + 2\boldsymbol{K}\boldsymbol{\theta} \right)$$

$\varepsilon > 0$:ステップ幅

劣微分の計算

$$t_j = \partial_{\theta_j} \max\Big(0, 1 - f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) y_i\Big)$$

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{j=1}^{n} \theta_j K(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

- $1 - f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) y_i > 0$ のとき $t_j = -y_i K(\boldsymbol{x}_i, \boldsymbol{x}_j)$
- $1 - f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) y_i = 0$ かつ $y_i = +1$ のとき

  $$t_j \in [-K(\boldsymbol{x}_i, \boldsymbol{x}_j), 0]$$
- $1 - f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) y_i = 0$ かつ $y_i = -1$ のとき

  $$t_j \in [0, K(\boldsymbol{x}_i, \boldsymbol{x}_j)]$$
- $1 - f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) y_i < 0$ のとき $t_j = 0$

劣勾配法による実装

$$\boldsymbol{\theta} \longleftarrow \boldsymbol{\theta} - \varepsilon \left(C \sum_{i=1}^{n} \partial_{\boldsymbol{\theta}} \max\Big(0, 1 - f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) y_i\Big) + 2\boldsymbol{K}\boldsymbol{\theta} \right)$$

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{j=1}^{n} \theta_j K(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

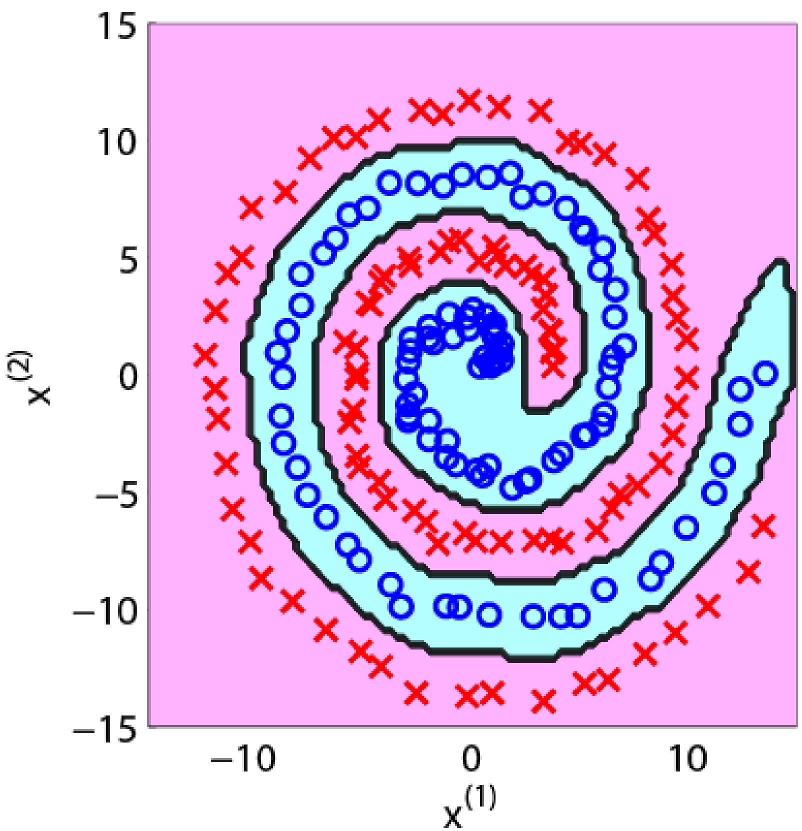
$\varepsilon > 0$:ステップ幅

- 劣勾配:

$$1 - f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) y_i > 0 \quad \Longrightarrow \quad \partial_{\theta_j} \max\Big(0, 1 - f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) y_i\Big) = -y_i K(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

$$1 - f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) y_i \leq 0 \quad \Longrightarrow \quad \partial_{\theta_j} \max\Big(0, 1 - f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) y_i\Big) = 0$$

実行例

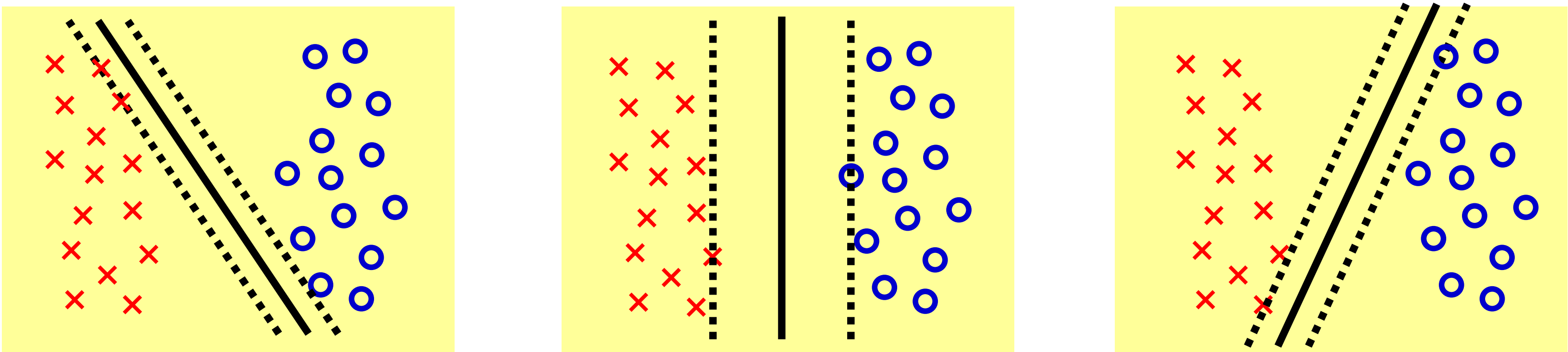


$$f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) = \sum_{j=1}^{n} \theta_j \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2h^2}\right)$$

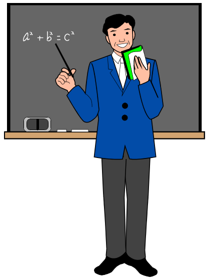
実装は宿題

- 複雑なデータもうまく分類できる

分類のまとめ



- 適切な分離平面を構成することでラベルを識別
- サポートベクトルマシン: マージン最大化
- カーネルトリックにより容易に非線形化可能



講義の流れ

1. 回帰
2. 分類

まとめ

- 教師付き学習：訓練データ（入出力の組）から，その背後に潜む関数を学習
- 回帰も分類も「損失＋正則化(regularization)」の最小化

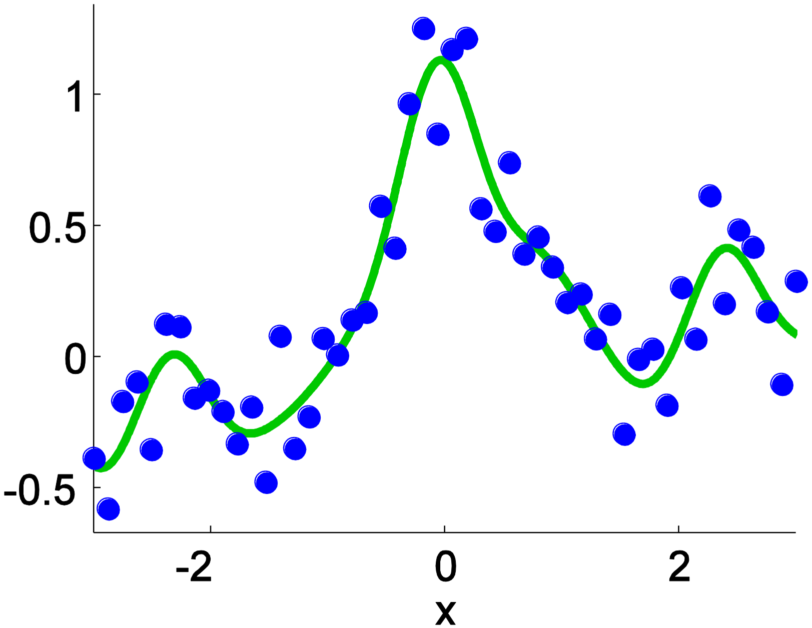
$$\min_{\boldsymbol{\theta}} \left[\sum_{i=1}^{n} \mathrm{loss}\Big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), y_i \Big) + \lambda \mathrm{Reg}(\boldsymbol{\theta}) \right]$$

宿題1

- ガウスカーネルモデルに対する正則化最小二乗回帰に含まれるバンド幅 h と正則化パラメータ λ を交差確認法によって決定せよ

$$\min_{\boldsymbol{\theta}} \left[\frac{1}{2} \sum_{i=1}^{n} \Big(y_i - f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) \Big)^2 + \frac{\lambda}{2} \sum_{j=1}^{n} \theta_j^2 \right]$$

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{j=1}^{n} \theta_j \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}_j\|^2}{2h^2} \right)$$



宿題1（続き）

```
clear all; rand('state',0); randn('state',0);
n=50; x=linspace(-3,3,n)'; pix=pi*x;
y=sin(pix)./(pix)+0.1*x+0.2*randn(n,1);

ガウス関数の幅hh=2*h^2と正則化パラメータlを
交差確認法によって決める;
x2=x.^2;
k=exp(-(repmat(x2,1,n)+repmat(x2',n,1)-2*x*x')/hh);
t=(k^2+l*eye(n))\(k*y);

N=1000; X=linspace(-3,3,N)';
X2=X.^2;
K=exp(-(repmat(X2,1,n)+repmat(x2',N,1)-2*X*x')/hh);
F=K*t;
figure(1); clf; hold on; axis([-2.8 2.8 -1 1.5]);
plot(X,F,'g-'); plot(x,y,'bo');
```

宿題1（続き）

- (MATLAB/Octave以外の場合)
  データは以下の規則で生成したものを用いる

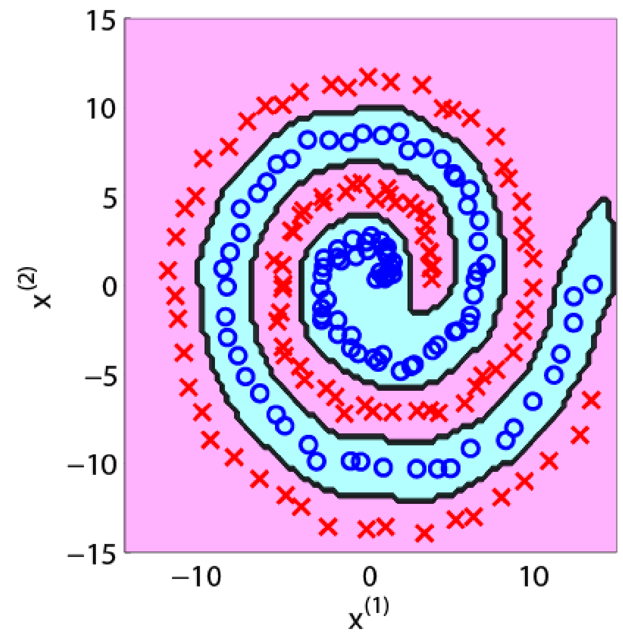
$$x_i = 6\left(\frac{i-1}{n-1} - \frac{1}{2}\right), \quad i \in \{1, 2, \ldots, n\}$$

$$y_i = \frac{\sin(\pi x_i)}{\pi x_i} + 0.1 x_i + \epsilon_i$$

$$\epsilon_i \sim N(0, 0.2^2)$$

宿題2

- ガウスカーネルに対する
サポートベクトルマシンの
劣勾配アルゴリズムを実装せよ



$$f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) = \sum_{j=1}^{n} \theta_j \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}_j\|^2}{2h^2}\right)$$

- 劣勾配

$$1 - f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)y_i > 0 \quad \Rightarrow \quad \partial_{\theta_j} \max\left(0, 1 - f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)y_i\right) = -y_i \exp\left(-\frac{\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2}{2h^2}\right)$$

$$1 - f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)y_i \leq 0 \quad \Rightarrow \quad \partial_{\theta_j} \max\left(0, 1 - f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)y_i\right) = 0$$

宿題2（続き）


```
clear all; rand('state',0); randn('state',0);
n=200; a=linspace(0,4*pi,n/2);
u=[a.*cos(a) (a+pi).*cos(a)]'+rand(n,1);
v=[a.*sin(a) (a+pi).*sin(a)]'+rand(n,1);
x=[u v]; y=[ones(1,n/2) -ones(1,n/2)]';

C=1; hh=2*1^2;
t=SVM(x,y,C,hh);

m=100; X=linspace(-15,15,m)'; X2=X.^2;
U=exp(-(repmat(u.^2,1,m)+repmat(X2',n,1)-2*u*X')/hh);
V=exp(-(repmat(v.^2,1,m)+repmat(X2',n,1)-2*v*X')/hh);
figure(1); clf; hold on; axis([-15 15 -15 15]);
contourf(X,X,sign(V'*(U.*repmat(t,1,m))));
plot(x(y==1,1),x(y==1,2),'bo');
plot(x(y==-1,1),x(y==-1,2),'rx');
colormap([1 0.7 1; 0.7 1 1]);
```

SVM.m
を実装せよ

宿題2(続き)

- (MATLAB/Octave以外の場合)
  データは以下の規則で生成したものを用いる

$$\begin{pmatrix} x_{i,1}^{(+)} \\ x_{i,2}^{(+)} \end{pmatrix} = \begin{pmatrix} a_i \cos a_i + \epsilon_{i,1}^{(+)} \\ a_i \sin a_i + \epsilon_{i,2}^{(+)} \end{pmatrix}$$

$$\begin{pmatrix} x_{i,1}^{(-)} \\ x_{i,2}^{(-)} \end{pmatrix} = \begin{pmatrix} (a_i + \pi) \cos a_i + \epsilon_{i,1}^{(-)} \\ (a_i + \pi) \sin a_i + \epsilon_{i,2}^{(-)} \end{pmatrix}$$

$$a_i = \frac{4(i-1)}{n-1}, \quad i \in \{1, 2, \ldots, n\}$$

$$\epsilon_{i,j}^{(+)}, \epsilon_{i,j}^{(-)} \sim U(0,1)$$