サポートベクトルマシン

6/1 ゼミ

加藤万理子

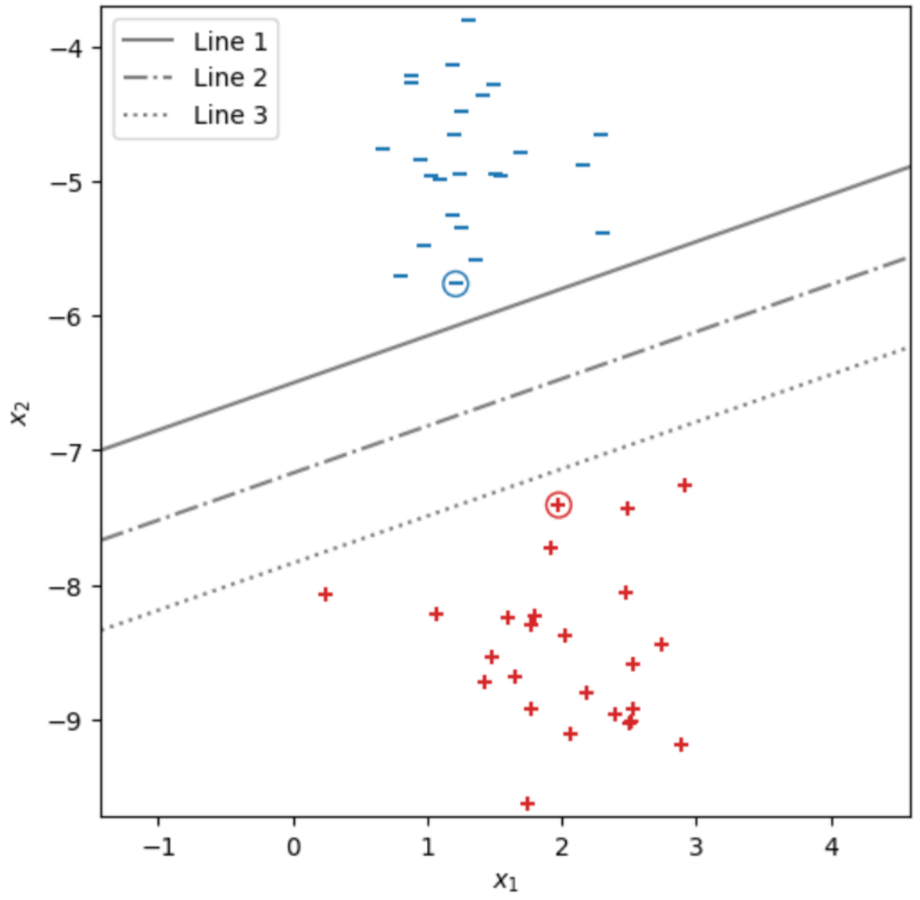
サポートベクトルマシン

- **分類を行うモデル**のひとつ
- 今回は説明として２値分類について述べる
- 回帰や多クラス分類にも使える

サポートベクトルマシン

ちょうど真ん中を通る直線を引きたい

→「点と直線の距離」の式から、最適な直線を導く



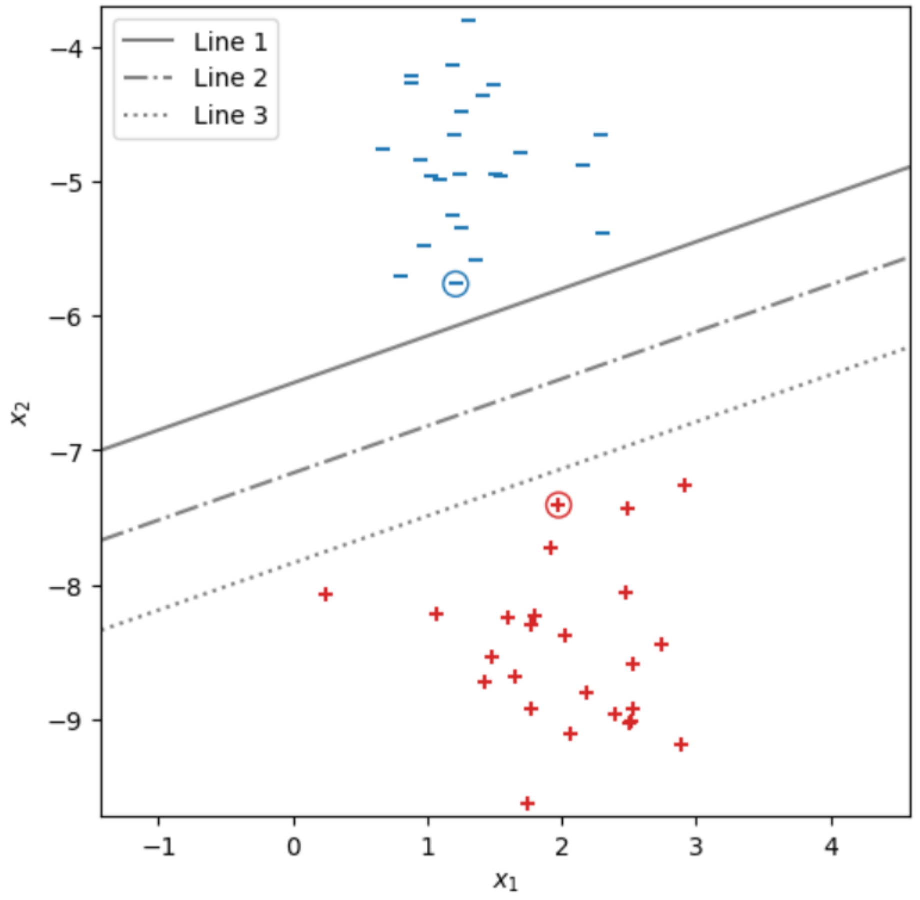
サポートベクトルマシン

- 直線で分離が可能

→一番近いデータまでの距離をハードマージンという

- 直線では分離が不可能

→誤判定を許す幅をソフトマージンという



定義

訓練集合：$\left\{ \left(x_i, y_i \right) \right\}_{i \in [n]}$

説明変数：$x_i \in \mathbb{R}^d$

ラベル：$y_i \in \{-1,1\}$

決定関数：$f(x) = \beta^T \boldsymbol{x} + b \qquad f: \mathbb{R}^d \to \mathbb{R}$

分類器：$g(x) = \begin{cases} 1 & f(x) > 0 \\ -1 & f(x) < 0 \end{cases}$

つなぎ

- $g(x)$の定義より、$f(x) = 0$が境目になっている。

→分類境界

- 正しく分類できているとき$y_i f\left(x_i\right) \geq 0$ がすべてのiで成立することに注意する

w, bを推定していく

点と直線の距離

命題23 より、平面$\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p = 0$と$x = [x_1, \cdots, x_p]$として、

$$d(x) := \frac{\left|\beta_0 + \ x_1\beta_1 + \cdots + x_p\beta_p\right|}{\sqrt{\beta_1^2 + \cdots + \beta_p^2}}$$

となる。

点と直線の距離

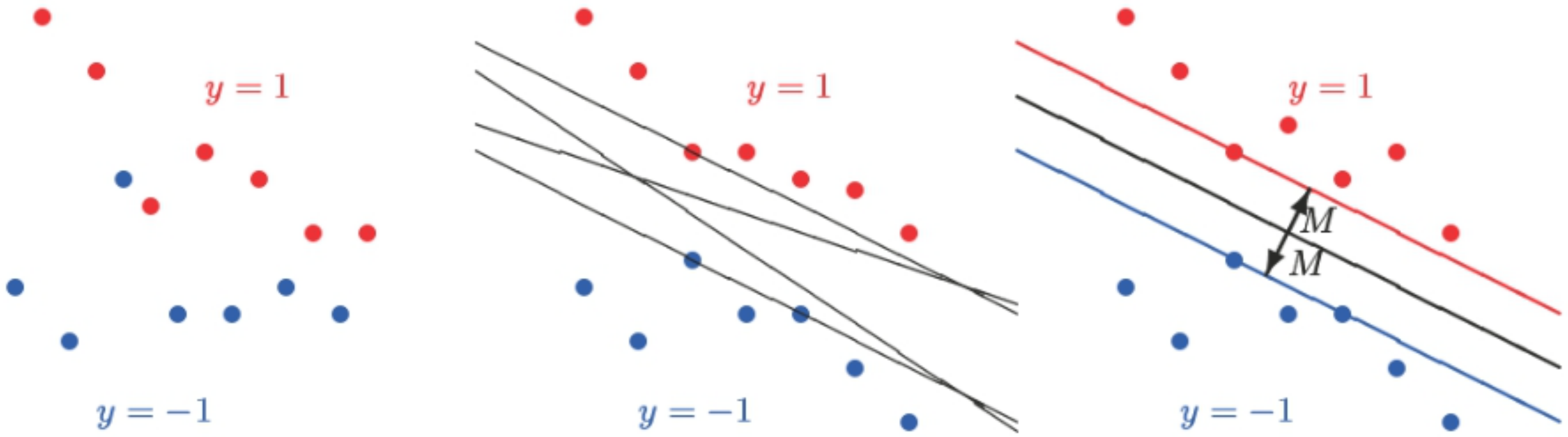
$$d(x) := \frac{\left|\beta_0 + \ x_1\beta_1 + \cdots + x_p\beta_p\right|}{\sqrt{\beta_1^2 + \cdots + \beta_p^2}}$$

あるx_iから分類境界までの距離は

$$\frac{\left|\beta^T \boldsymbol{x}_i + b\right|}{\left\|\beta\right\|}$$

である。

マージン最適化の表現



マージン最適化の表現

ある正の実数$M > 0$に対して$y_i f\left(x_i\right) \geq M$がすべての$i$で成立するとする。

$$\max_{\beta, b, M} \frac{y_i\left(\beta^T \boldsymbol{x}_i + b\right)}{\left\|\beta\right\|}$$

$$s.t. \quad y_i\left(\beta^T \boldsymbol{x}_i + b\right) \geq M, \quad i \in [n]$$

マージン最適化の表現

最大化するには$\boldsymbol{M}$はすべての事例に対する制約条件の値のうち、

最小の値とおなじになる。そのようなiをi'とすると、

$$\frac{\boldsymbol{M}}{\|\beta\|} = \frac{y_{i'}\left(\beta^T \boldsymbol{x}_{i'} + b\right)}{\|\beta\|}$$

マージン最適化の表現

$\left\|\beta\right\|_2 = 1$であることを仮定すると、

$$\max_{i=1,\cdots,N} y_{i'}\left(\beta^T \boldsymbol{x}_{i'} + b\right)$$

を求める問題となる。

ソフトマージン

分離不可では制約条件$y_i\left(\beta^T \boldsymbol{x}_i + b\right) \geq \boldsymbol{M}$を緩和することで導かれる。

$$\frac{y_i\left(\beta^T \boldsymbol{x}_i + b\right)}{\boldsymbol{M}} \geq 1$$

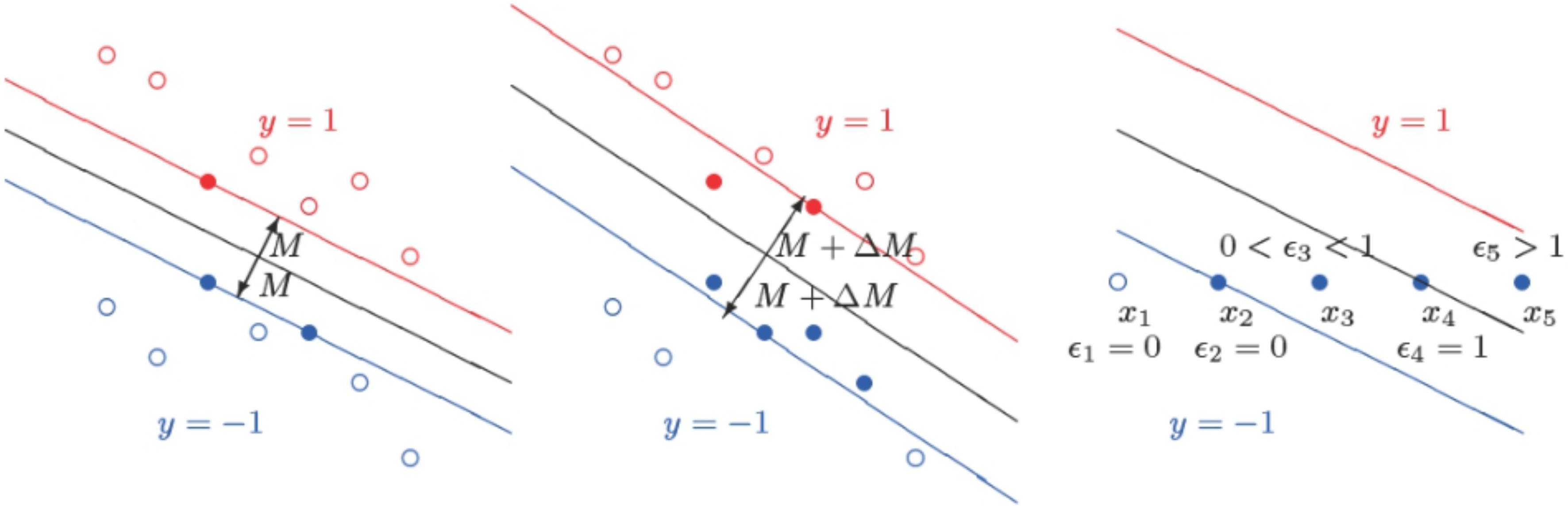
ここで新たに変数$\varepsilon_i \geq 0,\ i \in [n]$を導入する。

$$\frac{y_i\left(\beta^T \boldsymbol{x}_i + b\right)}{\boldsymbol{M}} \geq 1 - \varepsilon_i$$

$$y_i\left(\beta^T \boldsymbol{x}_i + b\right) \geq \boldsymbol{M}\left(1 - \varepsilon_i\right)$$

ソフトマージン

マージンと別に変数を導入することで、以下のように分類境界から外れているデータについても考えることができる。



ソフトマージンでの誤分類

誤分類が起こるとき、

$$y_i\left(\beta^T \boldsymbol{x}_i + b\right) < 0$$

より、以下である必要がある。

$$\varepsilon_i > 1$$

このことから、ある整数γについて

$$\sum_{i \in [n]} \varepsilon_i \leq \gamma$$

であれば、誤分類の数もγ以下であるとわかる。

主問題（ソフトマージン）

$$\min_{\varepsilon,\beta,b} \frac{1}{2}\left\|\beta\right\|^2 + C \sum_{i\in[n]} \varepsilon_i$$

$$s.t. \quad y_i\left(\beta^T x_i + b\right) \geq 1 - \varepsilon_i,$$

$$\mathrm{i} \in [n], \;\; \varepsilon_i \geq 0$$

Cは正則化係数と呼ばれる正の係数で、事前に値を決めるもの。

ラグランジュ関数

$$L_p = \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{N}\epsilon_i - \sum_{i=1}^{N}\alpha_i\left\{y_i(\beta_0 + x_i\beta) - (1 - \epsilon_i)\right\} - \sum_{i=1}^{N}\mu_i\epsilon_i$$

α_i, μ_iはラグランジュ係数/双対係数という。 $\left\|\beta\right\|$を最小にすることは、 $\boldsymbol{M}$を最大化することに相当する。

主問題と双対問題

ラグランジュ関数について、以下が主問題と等価である。

$$\min_{\beta,b,\varepsilon}\max_{\alpha\geq 0,\mu\geq 0} L(\beta, b, \varepsilon, \alpha, \mu)$$

以下は双対問題と呼ばれる。

$$\max_{\alpha\geq 0,\mu\geq 0}\min_{\beta,b,\varepsilon} L(\beta, b, \varepsilon, \alpha, \mu)$$

主問題と双対問題

基本、最適化問題では、双対関数の値は、主問題の最適値を超えない

→SVMでは主問題と双対問題の目的関数値が最適解において一致

→双対問題のほうが解きやすかったり、非線形化を見据えられたりするのでこちらを解く

KKT条件(Karush-Kuhn-Tucker)

- 解の最適性を判定するための条件
- 不等式を含む最適化問題を解くときに使う
- SVMではKKT条件が必要十分条件である

定義

シンプルなラグランジュ関数を定義する。

まず、想定する問題は

$$\min_{\beta} f_0(\beta)$$

$$s.t. \quad f_j(\beta) \le 0, \quad j = 1, \cdots, m$$

最適解がβ^*であるとする。このときラグランジュ関数は

$$L(\alpha, \beta) = f_0(\beta) + \sum_{j=1}^{m} \alpha_j f_j(\beta)$$

命題24 KKT条件

β^*を最適解とする。$f_1(\beta) \le 0, \cdots, f_m(\beta) \le 0$もとで、$\beta = \beta^* \in \mathbb{R}^p$が$f_0\left(\beta\right)$を最小にすることと、

$$f_1\left(\beta^*\right), \cdots, f_m\left(\beta^*\right) \le 0$$

であって、

$$\alpha_1 f_1\left(\beta^*\right) = \cdots = \alpha_m f_m\left(\beta^*\right) = 0$$

$$\nabla f_0\left(\beta^*\right) + \sum_{i=1}^{m} \alpha_i \nabla f_i\left(\beta^*\right) = 0$$

を満足する$\alpha_1, \cdots, \alpha_m \ge 0$が存在することは同値である。

KKT条件

必要十分の証明

双対問題

$$\max_{\alpha} -\frac{1}{2} \sum_{i,j \in [n]} \alpha_i \alpha_j y_i y_j x_i^T x_j + \sum_{i \in [n]} \alpha_i$$

$$s.t. \quad \sum_{i \in [n]} \alpha_i y_i = 0$$

$$0 \leq \alpha_i \leq C, \quad i \in [n]$$

命題25

$$\alpha_i = 0 \Longleftarrow y_i(\beta_0 + x_i\beta) > 1 \text{(外側)}$$

$$0 < \alpha < C \Longrightarrow y_i(\beta_0 + x_i\beta) = 1 \text{(上)}$$

$$\alpha_i = C \Longleftarrow y_i(\beta_0 + x_i\beta) < 1 \text{(内側)}$$

αからわかる位置関係

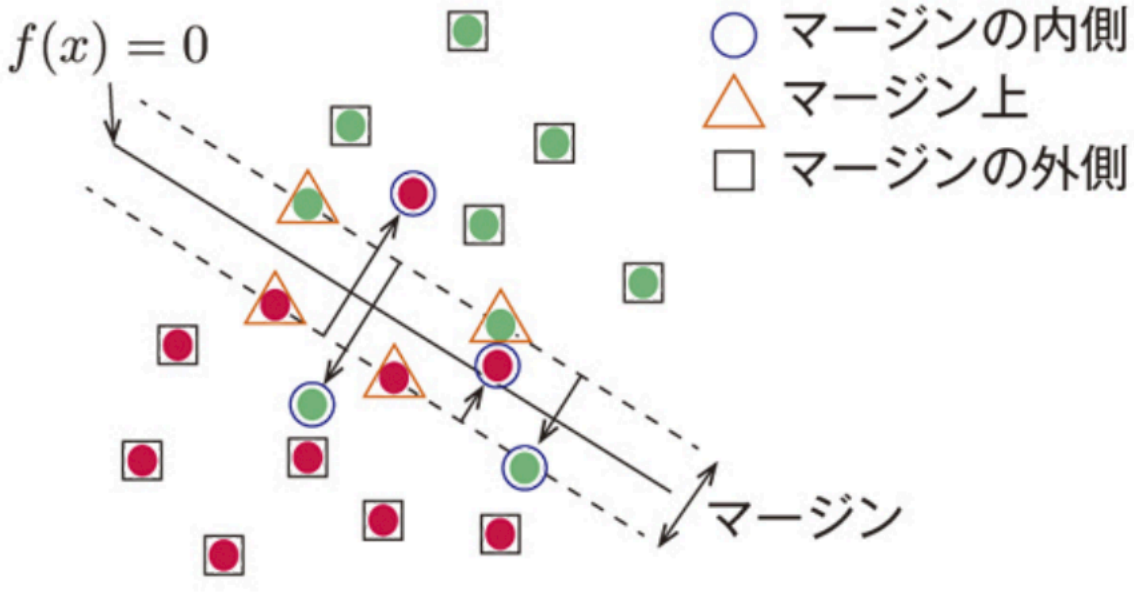


図 1.6　マージンの内側・マージン上・マージンの外側.

cvxoptを使ったSVMの実装

Pythonのcvxoptというライブラリを使う。cvxoptでは以下のような形の問題を想定している。

$$\min -\frac{1}{2}x^T P x + q^T x$$

$$s.t. \quad Gx \leq h$$

$$Ax = b$$

cvxoptを使ったSVMの実装

$$z = \begin{bmatrix} x_{1,1}y_1 & \cdots & x_{1,p}y_1 \\ \vdots & \ddots & \vdots \\ x_{N,1}y_N & \cdots & x_{N,p}y_N \end{bmatrix} \in \mathbb{R}^{N\times p}\ ,\ G = \begin{bmatrix} 1 & \cdots & 0 \\ 0 & \ddots & 0 \\ 0 & \cdots & 1 \\ -1 & \cdots & 0 \\ 0 & \ddots & 0 \\ 0 & \cdots & -1 \end{bmatrix} \in \mathbb{R}^{2N\times N}$$

cvxoptを使ったSVMの実装

$$P = -zz^T \in \mathbb{R}^{N\times N}$$

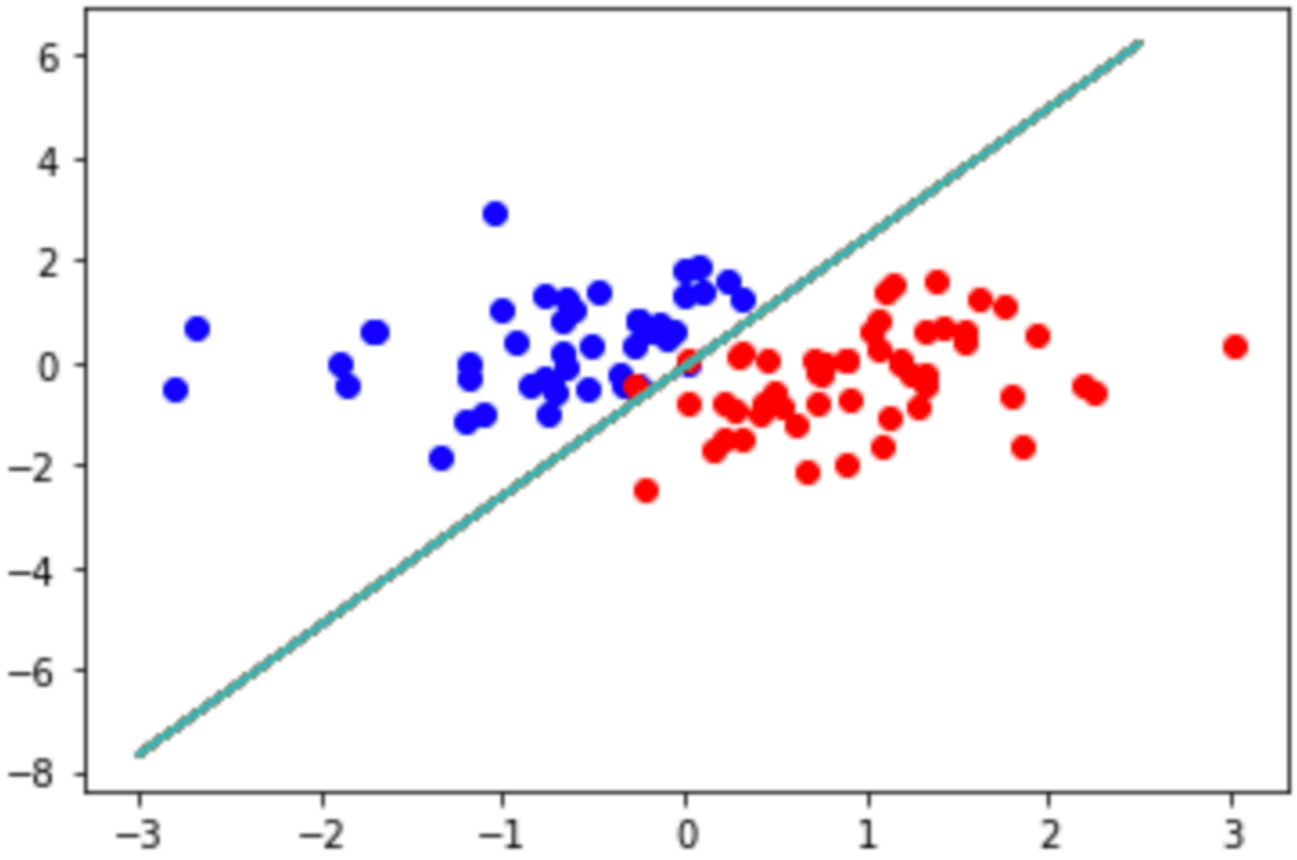
$$h = [C, \cdots, C, 0, \cdots, 0]^T \in \mathbb{R}^{2N+1}$$

$$x = [x_1, \cdots, x_N] \in \mathbb{R}^N$$

$$A = [-y_1, \cdots, -y_N] \in \mathbb{R}^{1\times N}$$

$$b \in \mathbb{R}$$

cvxoptを使ったSVMの実装



カーネル法を使った拡張

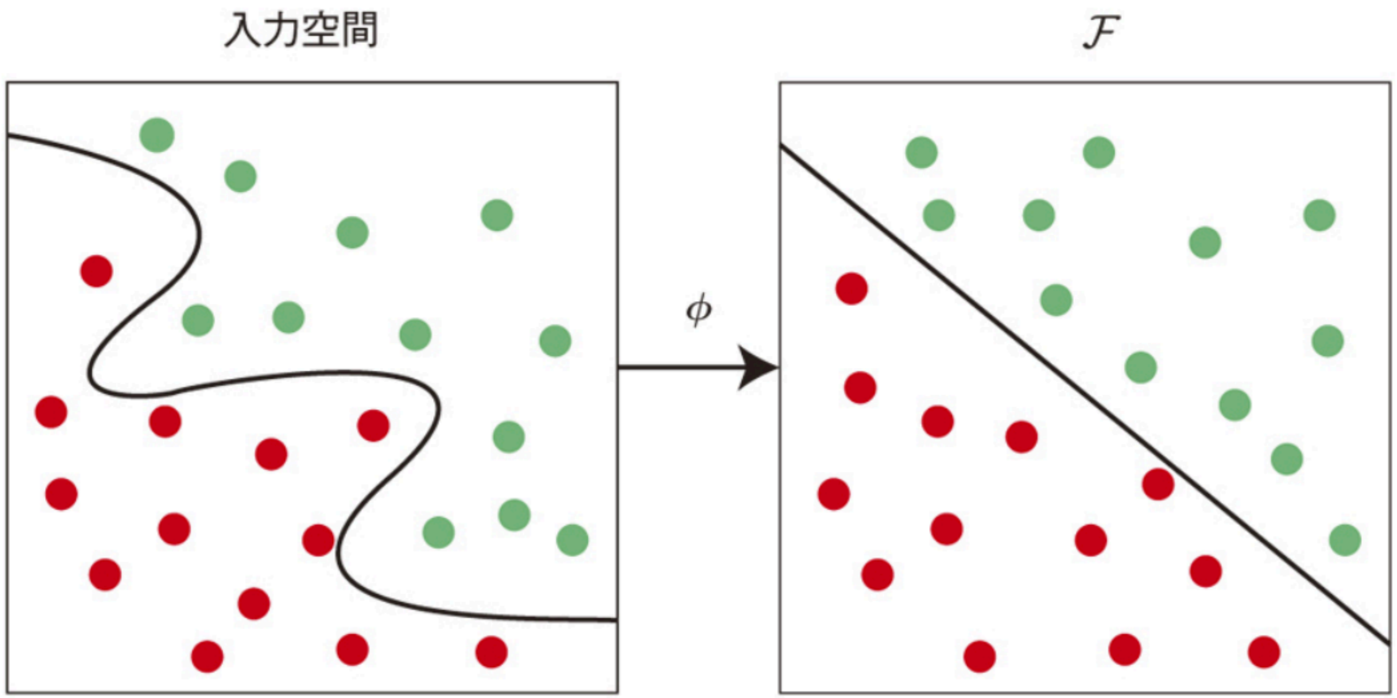
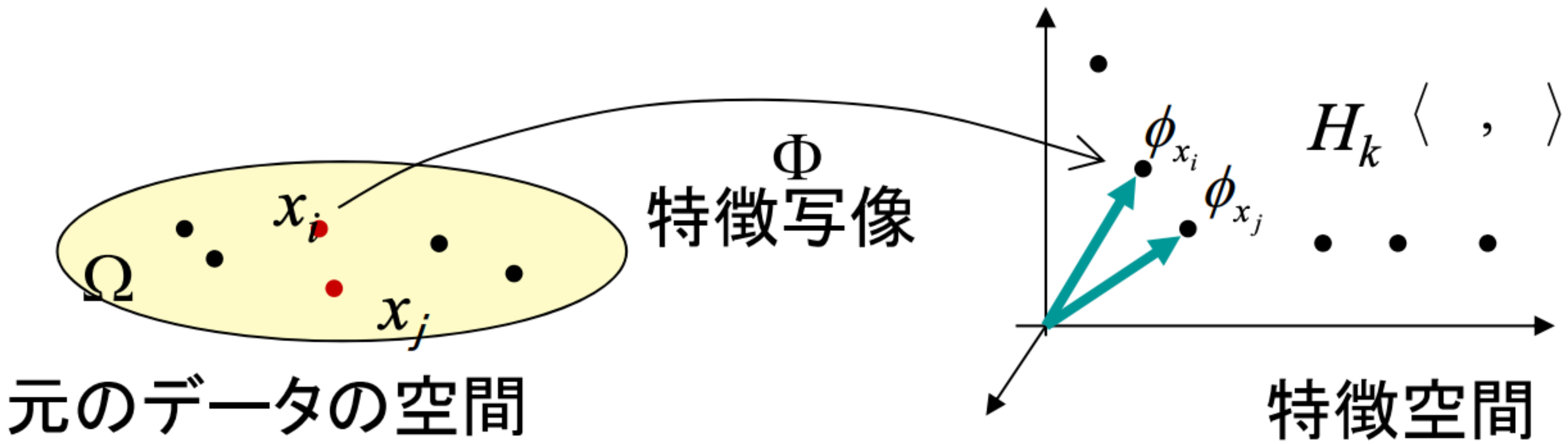


図 1.8　写像 $\boldsymbol{\phi}$ が非線形な場合，入力 $\boldsymbol{x}$ の空間（左）では決定関数 $f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x}) + b$ は非線形の分類境界を形成します．一方，$f(\boldsymbol{x})$ は $\boldsymbol{\phi}(\boldsymbol{x})$ に関しては 1 次なため $\boldsymbol{\phi}(\boldsymbol{x})$ の空間 $\mathcal{F}$（右）では線形な分類境界として表現されています．

カーネル法を使った拡張

– カーネル法の概念図



特徴空間で線形データ解析を施す!

e.g. SVM

カーネル法を使った拡張

$$f(x) = \boldsymbol{w}^T \boldsymbol{x} + b$$

について、入力xを何らかの特徴空間Fへ写像する関数を考える。

$$\phi : \mathbb{R}^d \to F$$

この$\phi(x)$を新たな特徴ベクトルと解釈すると、

$$f(x) = w^T \phi(x) + b$$

新たな特徴ベクトルを使い双対問題を考える

$$
\begin{aligned}
\max_{\boldsymbol{\alpha}} \quad & -\frac{1}{2} \sum_{i,j \in [n]} \alpha_i \alpha_j y_i y_j \boldsymbol{\phi}(\boldsymbol{x}_i)^\top \boldsymbol{\phi}(\boldsymbol{x}_j) + \sum_{i \in [n]} \alpha_i \\
\text{s.t.} \quad & \sum_{i \in [n]} \alpha_i y_i = 0 \\
& 0 \leq \alpha_i \leq C, \ i \in [n]
\end{aligned}
$$

カーネル関数の定義

注目すべきは$\phi(x_i)^T\phi(x_j)$という内積の形でのみであること。ここで、内積をカーネル関数として以下のように定義する。

$$K(x_i, x_j) = \phi(x_i)^T\phi(x_j)$$

内積は以下のような形でも表現できる。

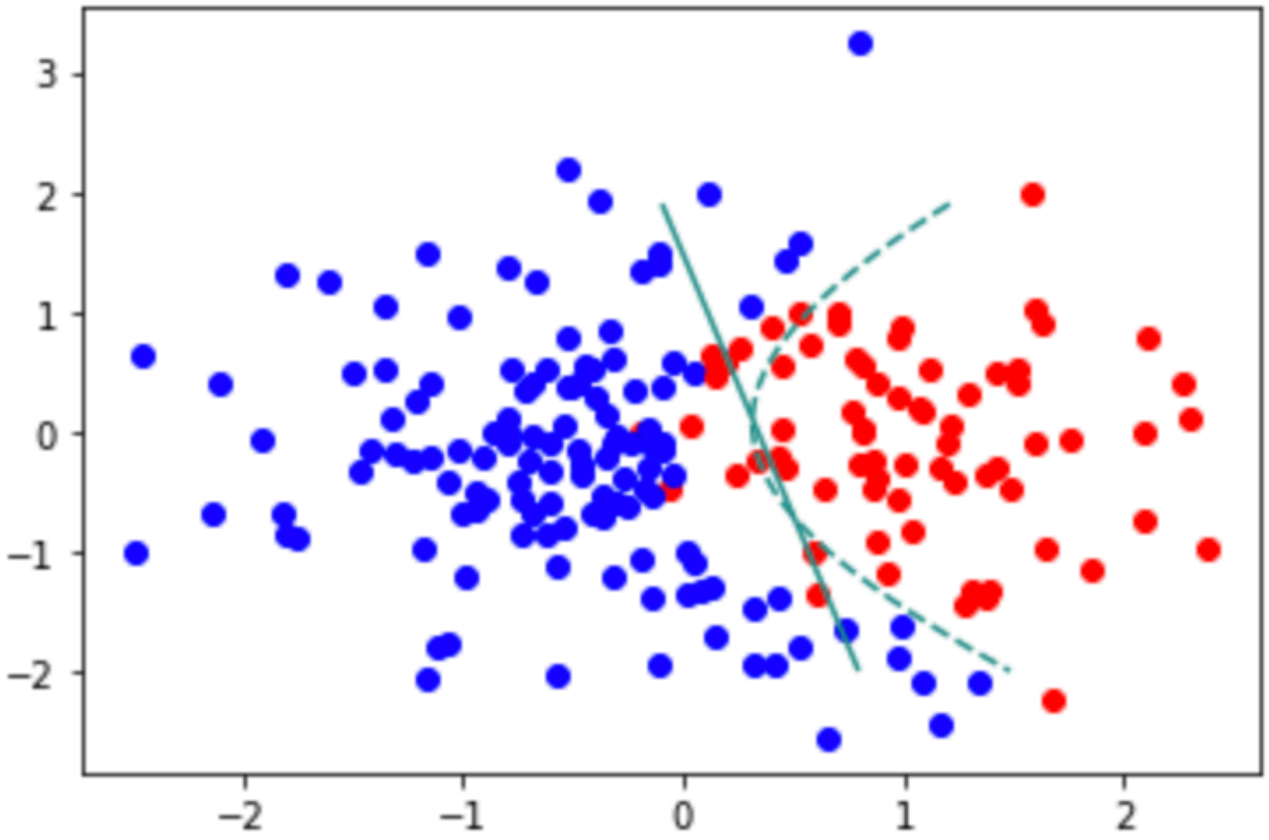
$$\langle \phi(x_i), \phi(x_j) \rangle$$

内積以外のカーネル

多項式カーネル(d次元)

$$K(x, y) = (1 + x^T y)^d$$

線形カーネルと非線形カーネルでの境界

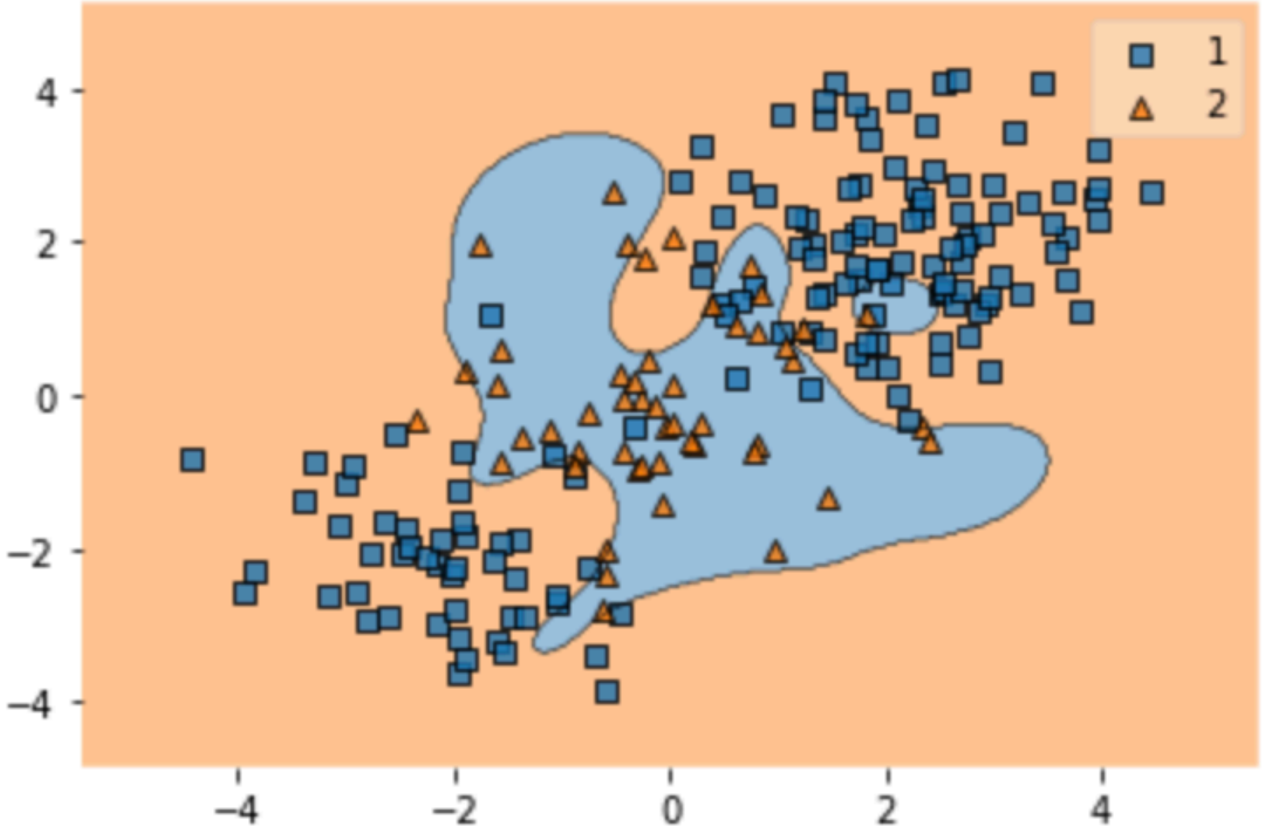


sklearnのSVMを用いる

Radial Basis Function(RBF/ガウスカーネル/ラジアルカーネル)

$$k(x, y) = \exp\left\{-\frac{1}{2\sigma^2} \| x - y \|^2\right\}$$

sklearnのSVMを用いる



ハイパーパラメーターチューニングを行う