サポートベクターマシン
Support Vector Machine
SVM

明治大学 理工学部 応用化学科
データ化学工学研究室　金子 弘昌

サポートベクターマシン (SVM) とは？

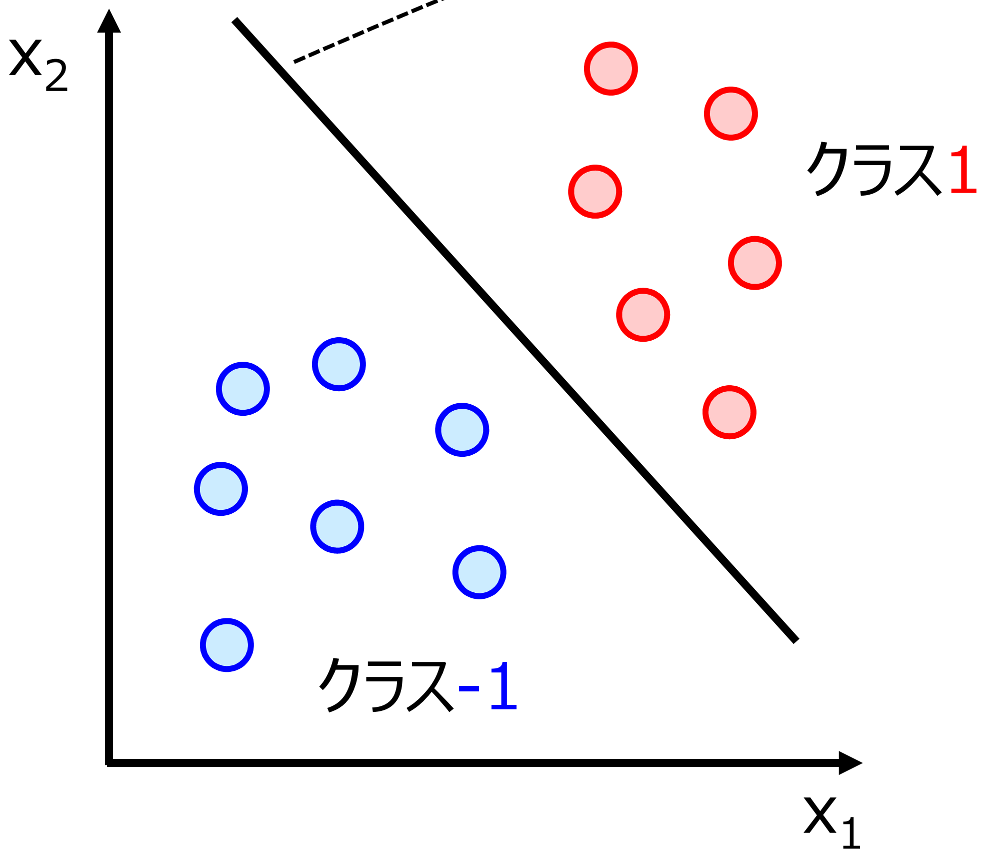
✓線形判別関数によるクラス分類

✓2つのクラス (1のクラス・-1のクラス) のどちらに属するか決定

✓予測能力の高いモデルを作成可能

✓カーネルトリックにより非線形の判別モデルに

線形判別関数

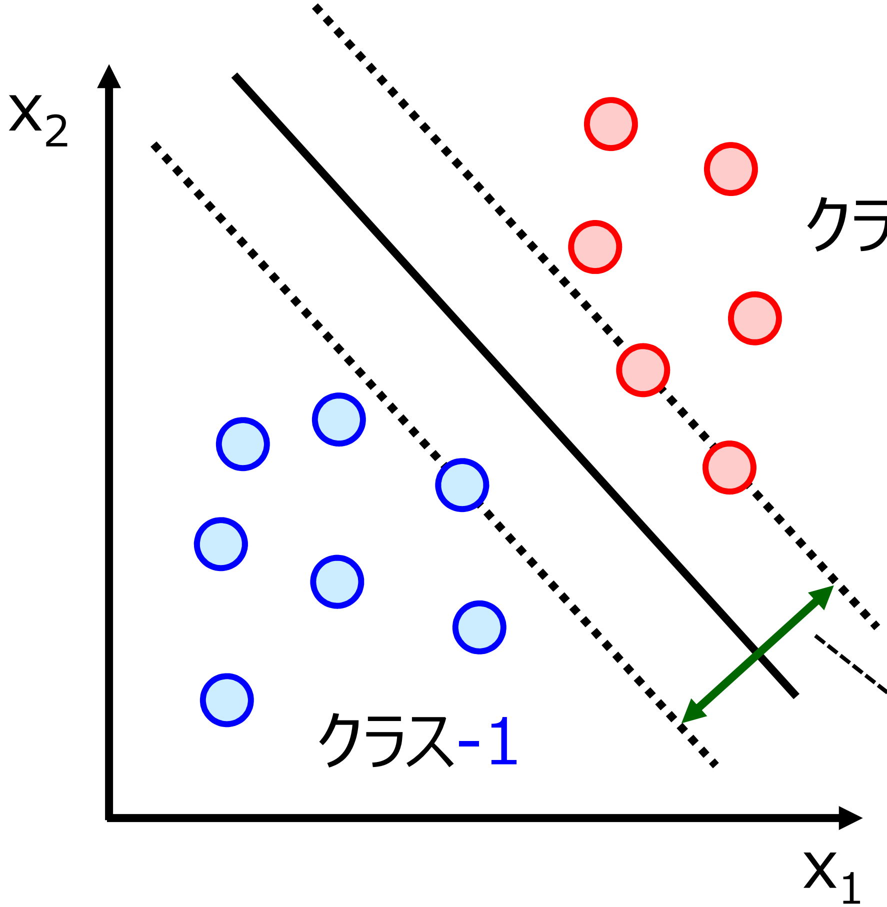


線形判別関数：

$$f(x_1, x_2) = w_1 x_1 + w_2 x_2 + b$$
$$= \mathbf{x}\mathbf{w} + b$$

$$\mathbf{x} = \begin{bmatrix} x_1 & x_2 \end{bmatrix},\ \mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$$

SVMの基本的な考え方



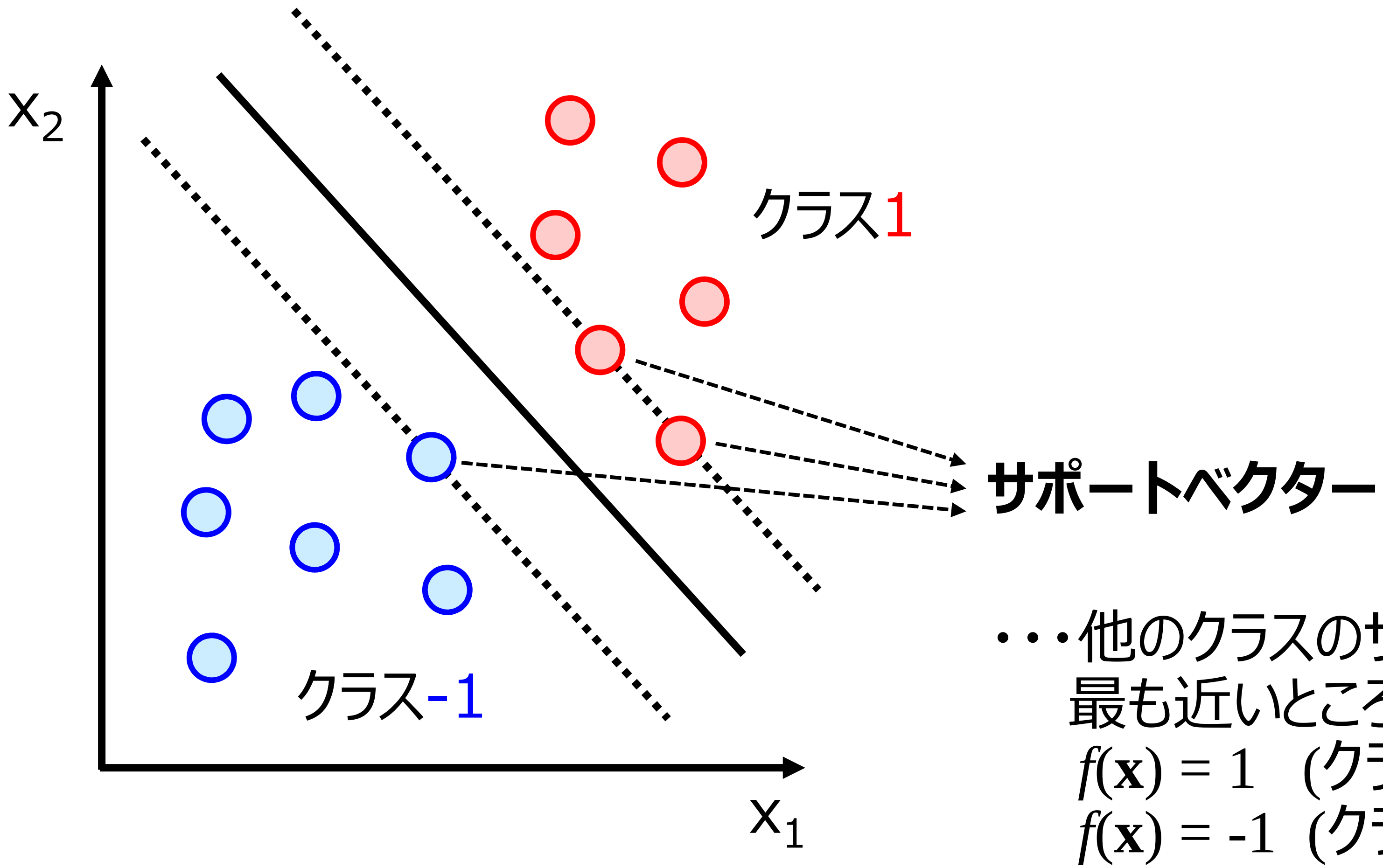
マージンを最大化するように判別関数を決める！

$$f(x_1, x_2) = w_1 x_1 + w_2 x_2 + b$$
$$= \mathbf{xw} + b$$

マージン $= \dfrac{2}{\|\mathbf{w}\|} = \dfrac{2}{\sqrt{w_1^{\ 2} + w_2^{\ 2}}}$

(点と直線との距離で計算)

サポートベクター

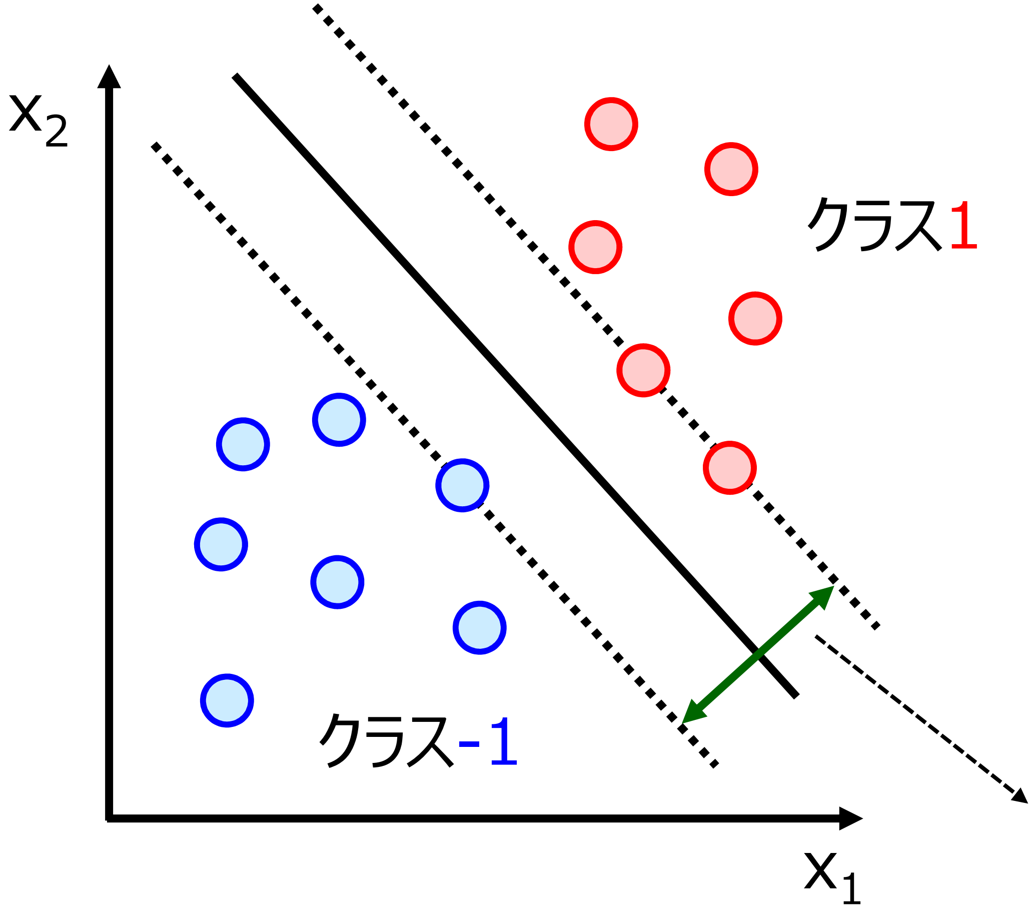


・・・他のクラスのサンプルと
最も近いところにいるサンプル

$f(\mathbf{x}) = 1$ (クラス1)

$f(\mathbf{x}) = -1$ (クラス-1)

マージンの最大化



線形判別関数：

$$f(x_1, x_2) = w_1 x_1 + w_2 x_2 + b$$
$$= \mathbf{xw} + b$$

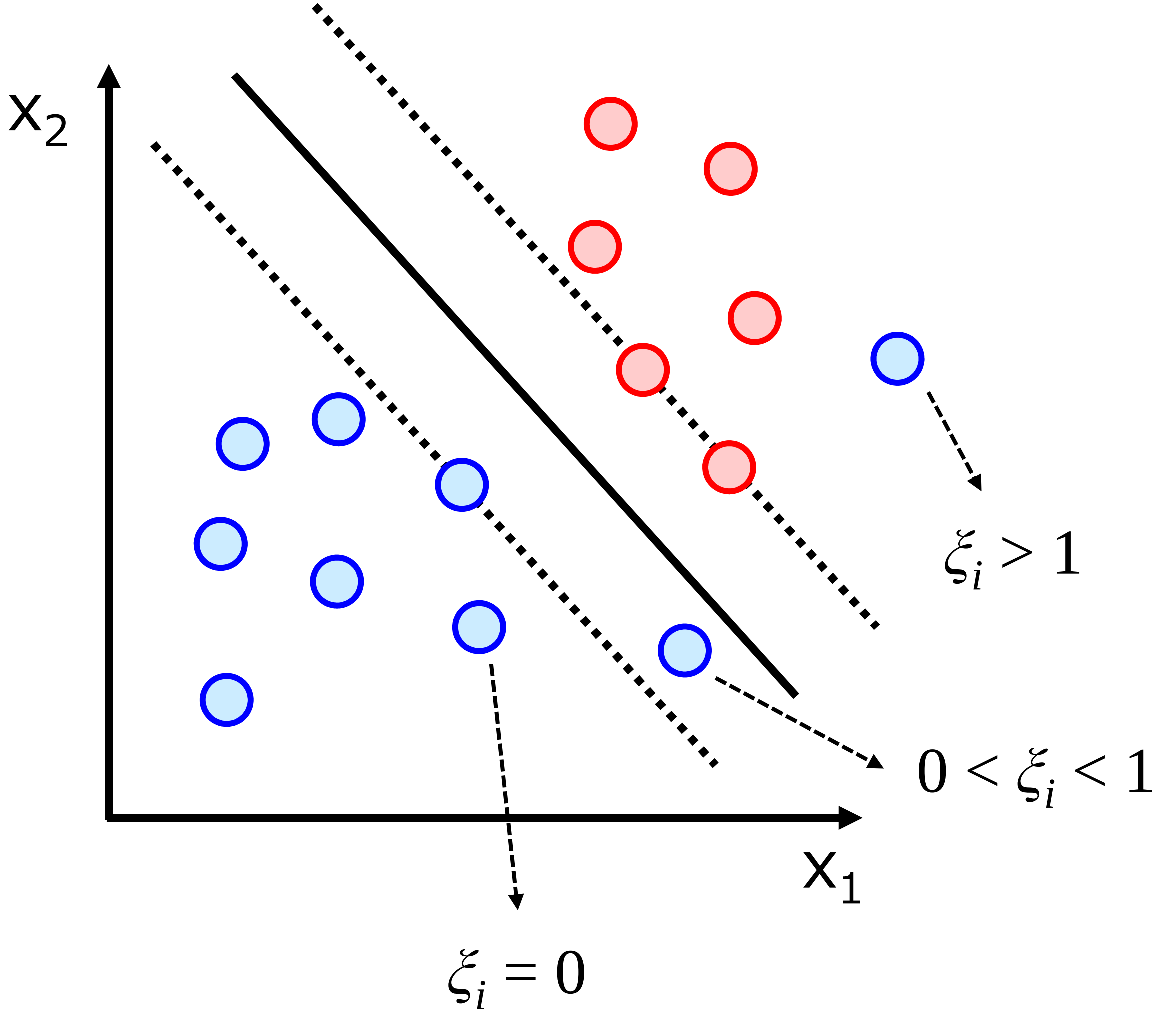
$$\frac{2}{\|\mathbf{w}\|}$$ の最大化

⇨ $$\frac{1}{2}\|\mathbf{w}\|$$ の最小化

マージン $= \dfrac{2}{\|\mathbf{w}\|} = \dfrac{2}{\sqrt{w_1^{\ 2} + w_2^{\ 2}}}$

(点と直線との距離で計算)

きれいに分離できないときは？



スラック変数 ξ を導入！

サンプルごとの ξ_i の和

$$\sum_{i=1}^{n} \xi_i$$

を最小化

n: モデル構築用サンプル数

２つの項を一緒に最小化

✓ $\|\mathbf{w}\| / 2$ の最小化 → 計算の都合上、$\|\mathbf{w}\|^2 / 2$ の最小化

✓ ξ_i の和の最小化

⇨ $$\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i$$ の最小化

ただし、 $\xi_i \geq 0, \quad y^{(i)} f\left(\mathbf{x}^{(i)}\right) \geq 1 - \xi_i$

C : ２つの項のバランスを決める係数
$\mathbf{x}^{(i)}$: i 番目のサンプルの説明変数
$y^{(i)}$: i 番目のサンプルの値 (1 もしくは -1)

重み w を求める

✓Lagrangeの未定乗数法

- ラグランジュ乗数α_i、β_i (i=1, 2, ..., n) を導入

$$G = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i\left\{y^{(i)} f\left(\mathbf{x}^{(i)}\right) - 1 + \xi_i\right\} - \sum_{i=1}^{n}\beta_i\xi_i$$

$\mathbf{w}$、b、ξ_iに関してGを最小化し、α_i、β_iに関してGを最大化

⇩

$\mathbf{w}$、b、ξ_iに関してGが極小

⇩

G を$\mathbf{w}$、b、ξ_i それぞれで偏微分して 0 とする

偏微分して0

G を $\mathbf{w}$ で偏微分して0

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y^{(i)} \mathbf{x}^{(i)\mathrm{T}}$$

G を b で偏微分して0

$$\sum_{i=1}^{n} \alpha_i y^{(i)} = 0$$

G を ξ_i で偏微分して0

$$\alpha_i + \beta_i = C \qquad (i = 1,\ 2,\ \ldots,\ n)$$

これらを使って
G を変形すると・・・

$$G = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y^{(i)} y^{(j)} \mathbf{x}^{(i)} \mathbf{x}^{(j)\mathrm{T}}$$

二次計画問題

$$G = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y^{(i)} y^{(j)} \mathbf{x}^{(i)} \mathbf{x}^{(j)\mathrm{T}}$$

制約　$0 \le \alpha_i \le C, \ \sum_{i=1}^{n} \alpha_i y^{(i)} = 0$　のもとで、

G を α_i に対して最大化する二次計画問題を解くと α_i が求まる

⇨ $\mathbf{w}$ が求まる　　$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y^{(i)} \mathbf{x}^{(i)\mathrm{T}}$

線形判別関数を求める

$$f(\mathbf{x}) = \mathbf{x}\mathbf{w} + b$$

$$= \mathbf{x}\sum_{i=1}^{n} \alpha_i y^{(i)} \mathbf{x}^{(i)\mathrm{T}} + b = \sum_{i=1}^{n} \alpha_i y^{(i)} \mathbf{x}\mathbf{x}^{(i)\mathrm{T}} + b$$

$$b = \frac{1}{n_{\mathrm{S}}} \sum_{i \in \mathrm{S}} \left(y^{(i)} - \sum_{j \in \mathrm{S}} \alpha_j y^{(i)} \mathbf{x}^{(i)} \mathbf{x}^{(j)\mathrm{T}} \right)$$

S : サポートベクター ($\alpha_i \neq 0$ のサンプル) の集合
n_{S} : サポートベクターの個数

非線形SVMへの拡張

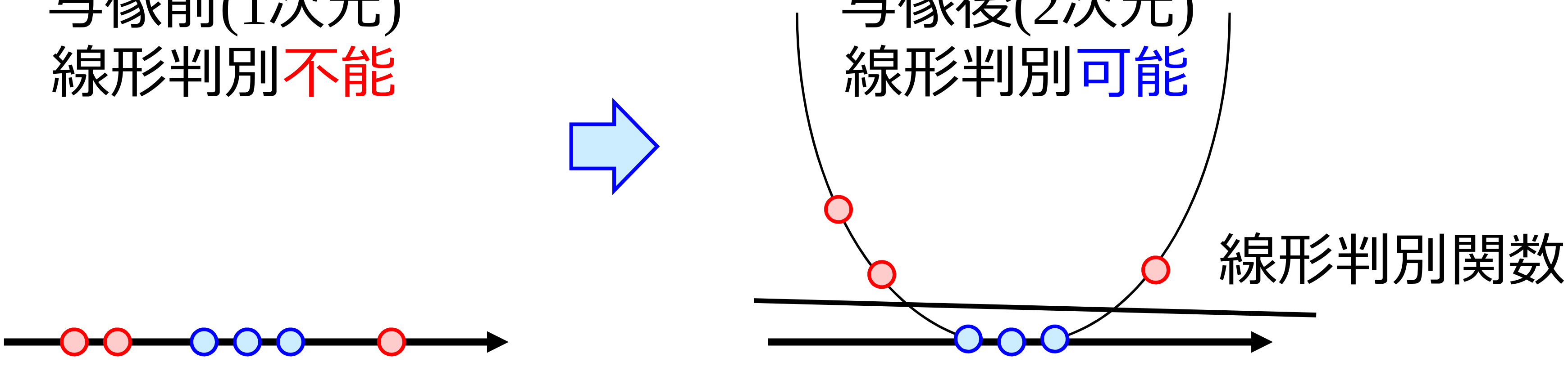
線形判別関数は判別能力に限界

- ✓ 元の空間より高次元に写像
- ✓ 高次元空間上で線形判別関数を構築

高次元空間への写像の例： $x \rightarrow \left(x,\ x^2 \right)$

写像前(1次元)
線形判別不能

写像後(2次元)
線形判別可能



カーネルトリック

線形判別関数 (元の空間)：　$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i y^{(i)} \mathbf{x}\mathbf{x}^{(i)\mathrm{T}} + b$

⇩ 高次元空間への写像：　$\mathbf{x} \rightarrow \phi(\mathbf{x})$

非線形判別関数 (高次元空間)：　$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i y^{(i)} \underline{\phi(\mathbf{x}) \phi\left(\mathbf{x}^{(i)}\right)^{\mathrm{T}}} + b$

$\phi(\mathbf{x})$ を求める必要はなく、内積 $\phi(\mathbf{x})\phi(\mathbf{x}^{(i)})^{\mathrm{T}}$ が分かればOK！

⇩

高次元空間への写像 $\phi(\mathbf{x})$ ではなく、内積を指定 (カーネル関数 K)

$$K\left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\right) = \phi\left(\mathbf{x}^{(i)}\right) \phi\left(\mathbf{x}^{(j)}\right)^{\mathrm{T}}$$

カーネル関数の例

✓線形カーネル

$$K\left(\mathbf{x}^{(i)},\mathbf{x}^{(j)}\right)=\mathbf{x}^{(i)}\mathbf{x}^{(j)\mathrm{T}}$$

✓ガウシアンカーネル (使われることが多い)

$$K\left(\mathbf{x}^{(i)},\mathbf{x}^{(j)}\right)=\exp\left(-\frac{\left\|\mathbf{x}^{(i)}-\mathbf{x}^{(j)}\right\|^2}{2\sigma^2}\right)=\exp\left(-\gamma\left\|\mathbf{x}^{(i)}-\mathbf{x}^{(j)}\right\|^2\right)$$

✓多項式カーネル

$$K\left(\mathbf{x}^{(i)},\mathbf{x}^{(j)}\right)=\left(1+\lambda\mathbf{x}^{(i)}\mathbf{x}^{(j)\mathrm{T}}\right)^d$$

最終的なSVMを作る前に最適化するパラメータ

✓ガウシアンカーネルを使用したSVMのとき、最初に C と γ とを適切に設定する必要がある

$$\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i$$

$$K\left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\right) = \exp\left(-\gamma\left\|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\right\|^2\right)$$

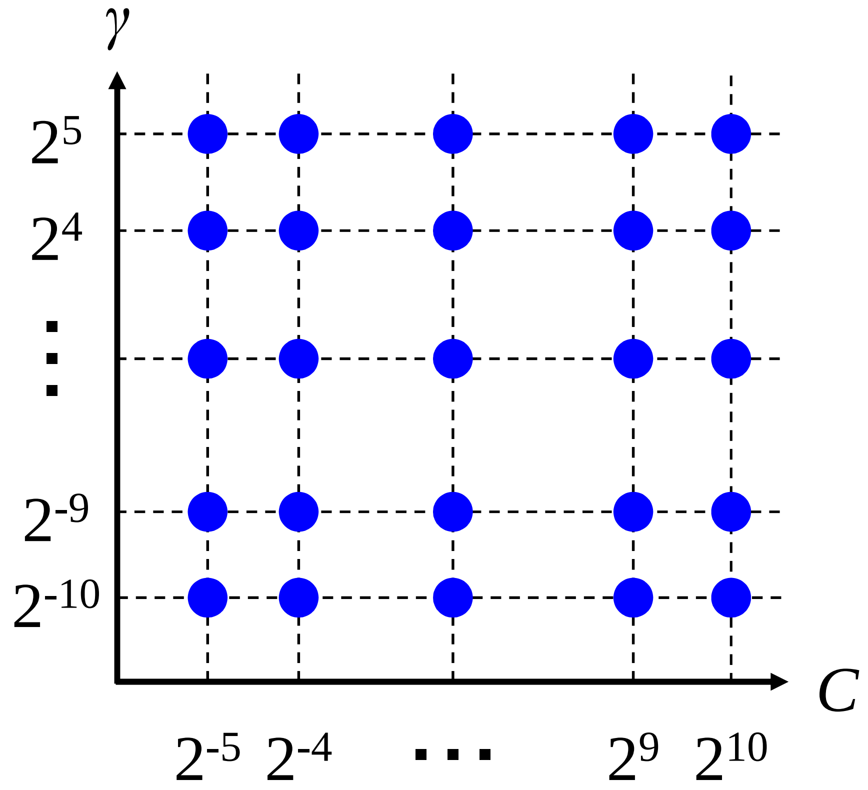
グリッドサーチ ＋ クロスバリデーション による C および γ の最適化

グリッドサーチ＋クロスバリデーション

✓C と γ の候補を設定し、すべての組合せ (グリッド, 下図の●) でクロスバリデーションを行う

✓C と γ の候補の例

- C : 2^{-5}, 2^{-4}, ..., 2^{9}, 2^{10}
- γ : 2^{-10}, 2^{-9}, ..., 2^{4}, 2^{5}



✓例) クロスバリデーション後の正解率が最も高い C・γ の組を選択

- 正解率については[こちら]